Implementation of Cache Supported Path Planning System

Akanksha Anil Gaikwad
Department of Computer Engineering
PES modern college of engineering

Prof. Dr. Mrs. S. A. Itkar
Department of Computer Engineering
PES Modern college of engineering

***Abstract*—Path planning is a very important function of road network navigation systems. Path planning is used to find out the route between the given starting location and destination location. Now a days path planning becomes very basic function in many mobile devices because of digital mapping of roads and high availability of global positioning systems(GPS). But there are some challenges in these systems for mobile users on roads, such as unexpected traffic conditions, sudden change in driving direction, unstable or lost GPS signals, and so on. In such cases, path planning service needs to be delivered in time to time fashion. When large number of path planning queries are submitted to the server then requirement of timeliness is more challenging. It is necessary for the server to handle heavy workload efficiently because response time is critical for user satisfaction. So, the Path Planning by Caching (PPC) system is used here to solve a new path planning query in real time by caching and reusing historical queried paths. PPC system also uses partially matched queries to find out some parts of the new query, where the conventional path planning systems uses only perfectly matching queries. So, in PPC server only needs to compute the unmatched part of query, thus overall system workload is reduced. Here, the system in this paper uses ppattern detection algorithm to detect the different possibilities of the shortest path and then by using shortest path estimation algorithm it find out the best path among them. Then cache construction and update mechanism reduces time required to find out shortest path. As a result, this system reduces overall computational latency.**

***Index Terms*—Path Planning, Spatial Database, Cache Management.**

I. Introduction

Way arranging is a fundamental capacity of street organize route benefits that finds a course Between the predetermined begin area and goal. Because of the wide accessibility of the Global situating framework and computerized mapping of streets, street organize route administrations have turned into an essential application on numerous cell phones. The proficient of this way arranging capacity is basic for portable clients on streets on account of different unique cases, for example, a sudden alter in driving course, unforeseen traffic conditions, unsteady or lost GPS signals, thus on[1]. In these cases, the way arranging administration should be conveyed in a convenient manner. The necessity of convenience is much all the more difficult when a mind-boggling number of way planning questions is submitted to the server. Thus, the Path Planning by Caching (PPC) framework is utilized to answer another way arranging question continuously by proficiently reserving and reusing verifiable questioned ways[1]. PPC use the mostly coordinated inquiries to answer parts of the new question is not normal for the traditional store based way arranging frameworks, where a questioned way in reserve is utilized just when it coordinates superbly with the newinquiry. In this way, Because of that server just needs to register the unmatched way fragments, accordingly fundamentally diminishing the general framework outstanding burden[1]. Way arranging by reserving (PPC), to effectively answer another way arranging question by using stored ways to abstain from experiencing a tedious most limited way calculation. The idea of Pattern that is a stored way which imparts portions to different ways. PPC underpins incomplete hits between P Patterns and another question. A tale probabilistic model is proposed to recognize the reserved ways that are of high likelihood to be a PPattern for the new question dependent on the coherency property of the street systems[4]. Another store substitution component is utilized by considering the client inclination among streets of different sorts. An ease of use measure is allotted for each question by tending to both the street type and inquiry notoriety to gauge the Path Planning Concepts.

A. *MOTIVATION*

Road network navigation becomes very popular application in many mobile devices. Path planning used to find out best path between starting location and destination point. Path planning service should be delivered to the users in time to time manner. It get the Current Location of Searching user and Get path from current Location.

B. *OBJECTIVE*

To Find out the Shortest Path between Current Location and Search point in real time by using path planning by caching system. This system is to find out shortest path by using historical queried paths from cache to avoid undergoing a time consuming shortest path computation.

II. REVIEW OF LITERATURE

This section reviews the research work related to the path planning system and shortest path estimation, which are similar to proposed system.

Path Planning: Path planning by caching is used to answer a path planning query by using cached historical paths to avoid a time consuming shortest path computation[1]. The idea of PPatterns is a path in a cache which shares part with other paths. PPC system also uses partial hits between PPatterns and a user's query. A probabilistic model is given to find out the cached paths that are of high probability to be a PPattern for the user's query based on the coherency property of the road networks[4]. Cache replacement policy is used for cache management. A usability measure is assigned to each query based on the road type and query popularity[3],[7]. For user's satisfaction the path planning service needs to be delivered in a real time. when an large number of path planning queries are given to the server then requirement of timeliness is more challenging. In the conventional path planning systems a path in cache is used only when it perfectly matches with the query given by user. PPC uses the partially matched queries to answer parts of the new query. So, Because of that server only needs to compute the unmatched part of query and it reduces the overall system workload[1].

Shortest path caching (SPC) system is also used for path planning[2]. SPC system uses the optimal sub path property. SPC system is cache based system that find out shortest path. In SPC system benefit value is assigned to a shortest path to determine whether to preserve that path in the cache or not. The benefit value of path is a sum of the benefit values of each sub path in that shortest path. Benefit value of a path p can be computed based on two factors: the Frequency of a path and its expense. The frequency of a path is computed based on the number of occurrences of the historical sub paths that overlaps path p. The expense of a path p is the computational time required for shortest path algorithm to find out shortest path p. The cache can be placed at either a proxy or the server. Cache is used to optimize the computation and communication costs at the serve and reduces the response time of shortest path queries. SPC system requires more time to find out shortest path as compare to PPC system. so its time complexity is more[2].

In hierarchical modeling of traffic flow with application to Malta's road network the route choice preferences between OD(origin-destination) pairs are estimated by using the Bayesian hierarchical model[3]. The method used in this system uses the origin-destination information as well as traffic counts of some links in the network to find out route preference probabilities. Multi-nominal distribution is used to represent the route choice preferences. Route choice preferences are estimated by using Markov Chain Monte Carlo(MCMC) algorithm[3]. In this system, given model takes into account three factors: the uncertainties presents in the origin-destination data, alternative route choice preferences and measurement errors in link counts both inside or outside the given network. In this system the Bayesian model and estimation techniques are used to solve traffic assignment problems. Dimensionality of the traffic assignment problem is limited by using the most commonly used route choice sets and still allowing the possibility of choosing alternative routes. A Bayesian model for such uncertainties, allows more detailed summaries of traffic counts of links and route choice preferences[3].

In the shared execution of path queries the headway of versatile innovations and the expansion of guide based applications have empowered a client to get to a wide assortment of administrations that run from data inquiries to route frameworks[4]. Because of the fame of guide based applications among the clients, the specialist organization regularly requires to answer a substantial number of synchronous questions. In this way, handling questions productively on spatial systems street have turned into an essential research territory as of late. In this system, center around way inquiries that locate the most brief way between a source and a goal of the client. Specifically, they address the issue of finding the most brief ways for countless way questions in street systems. Customary frameworks that think of one as inquiry at once are not reasonable for some applications because of high computational and benefit costs. These frameworks can't ensure required reaction time in high load conditions. This system proposes a productively aggregate based methodology that furnishes a down to earth arrangement with diminished expense. The key idea for this methodology is to gather inquiries that share a typical travel way and afterward register the most limited way for the gathering[4].

A* shortest path algorithm is used study the point-to-point shortest path problem in a setting where pre-processing is allowed[5]. This system improve the reach-based approach of Gutman in several ways[6]. In particular, it introduce a bidirectional version of the algorithm that uses implicit lower bounds and it add shortcut arcs to reduce vertex reaches. This modifications greatly improve both pre-processing and query times. The resulting algorithm is as fast as the best previous method, due to Sanders and Schultes. However, this algorithm is simpler and combines in a natural way with A* search, which yields significantly better query times. Finding shortest paths is a fundamental problem. In this paper focus is on road networks. However, this algorithms do not use any domain-specific information, such as geographical coordinates, and therefore can be applied to any network[5].

Reach based routing method is also used for shortest path finding[6]. Past work has uses two techniques for high volume shortest path searching on large graph such as a road network. One technique, extensively reached by the academic community. It pre-compute paths and avoid the too expensive all pair computations by computing shortest path and sorting only a few paths for an arbitrary origin and destination can be formed by joining a small number of pre-computed paths. This method implementation can be somewhat complex. A new concept called a reach allows to shortest path computation speed par industry approach[6].

HiTi Model: In this model they built up a HiTi (Hierarchical MulTi) chart show for organizing substantial land guides to accelerate the base cost course calculation[7]. The HiTi chart show gives a way to deal with abstracting and organizing a geological guide in a various leveled design. They propose another most brief way calculation named SPAH, which uses HiTi chart model of a land guide for its calculation. This system gives the verification for the optimality of SPAH. Execution investigation of SPAH on matrix diagrams shows that it essentially decreases the pursuit space over existing techniques. It system present an inside and out test examination of HiTi chart strategy by contrasting it and other comparative takes a shot at matrix diagrams. Inside the HiTi diagram structure, they used a parallel most brief way calculation called as ISPAH. Exploratory outcomes shows that entomb inquiry most brief way issue gives more chance to adaptable parallelism than the intra question briefest way issue[7].

Thus, the conventional systems requires more time to find out shortest path as compare to PPC system because these systems uses shortest path from cache only when it is perfectly matches with the user's query. Unlike the conventional systems PPC uses a path from cache even if it do not matches perfectly with the query given by user. PPC uses matched part of shortest path from cache and then compute unmatched part from server. So, PPC system reduces server workload and takes very less time to find out shortest path as compare to other systems.

III. SYSTEM ARCHITECTURE AND OVERVIEW

This system is used to estimate the shortest path between source location and destination location in real time by using shortest path estimation algorithm. Cache replacement policy is used for management of cache. PPattern detection algorithm is also used for detecting set of best matching patterns with user's query[1].

A. Architecture

1) The query is given to the system as an input. The query can be Place name, Location, Address.
2) The query contains source location and destination location. User gives query to the system server.
3) First query is given to the PPattern Detection algorithm. It detects the set of best matching patterns which matches with the new query.
4) The set of PPatterns is given to the shortest path estimation algorithm. It computes the best shortest path among them.

Explanation:

5) Users are first registered and use first login with OTP (one time password) and next time only login without OTP.

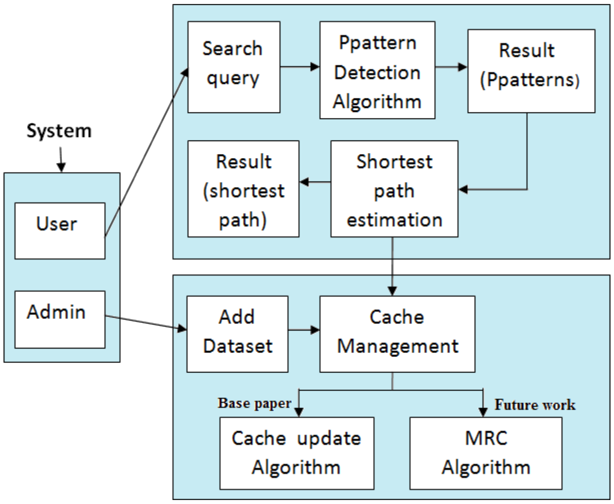


Fig. 1. System Architecture

6) After login users Search Location according to need then PPattern detection Algorithm find out set of best patterns. Result of PPattern Algorithm given to the Shortest Path Estimation module to get the Shortest Path of Search location.
7) To Improve the Speed of Application while Searching Cache Management System is Used.

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 Algorithm.

B. Algorithms

1) Cache Construction and Update Algorithm[1].

Input: a query, a cache C.
Output: Updated cache C with best shortest path.
Steps:

```
1. PT← Detect PPatterns.
2. p← Estimate shortest path from PT.
3. if Cache is not full then
4.     insert p into C;
5.     Return C.
6. else
7.     $\mu$ ← Calculate usability of each path in cache.
8.     p*← Minimum usability path.
9.     if $p*.\mu < p.\mu$ then
10.        C← Replace p* with p.
11.    End if.
12. End if.
13. Return C.
```

2) P Pattern Detection Algorithm[1].

Input: $q_{s,t}$:a query; D_l:Distance Threshold ; C:a cache; D_g:grid cell size.

Output: All candidate PPatterns PT.

Steps:

```
1. if $D(s, t) < D_l$ then
2. Return PT = 0.
3. End If.
4. Divide target space by grid size $D_g$.
5. Determine the start grid $g_s$ and destination grid $g_t$.
6. $Q_s$ ← Logged queries whose paths pass $g_s$.
7. $Q_t$ ← Logged queries whose paths pass $g_t$.
8. $Q = Intersect(Q_s, Q_t)$.
9. PT ← Sub paths from $tt_s$ to $tt_t$ for each query in Q.
10. End If.
11. Return PT.
```

3) Shortest path estimation Algorithm[1].

Input: A query source node s anjd Destination node tj; all candidate PPatterns PT; cache C.

Output: Estimated shortest path.

Steps:

```
1.PT← PPattern Detection.
1. if it Empty(PT) then
2. p* ← Calculate path from server and return.
3. end if.
5 Initialize estimated shortest distance $ESD = \infty$.
6. for each path p ∈ PT do
7. if p is a complete hit then
8. return $p_{s,t} = p$.
9. end if.
10. $s* = argmin_{s \in v_p} D(s\,,j\, s)$.
11. $d_s = D(sj,* s\,)$.
12. Remove s* from path node set.
13. $t* = argmin_{t \in v_P} D(t, tj)$.
14. $d_t = D(t*, t)$.
15. Let $d_r = |SDP\,(s\,j\,,j t|)$.
16. $\hat{d} = d_s + d_r + d_l$.
17. if $d < ESD$ then
18. $ESD = d$
19. Update best PPattern $P* = p_{s*,t*}$
20. end if.
21. end fo.r
22. if sj is not equal to $v_s^{p*}$ then
23. $SDP(sj, v_s^{p*})$ ← Compute shortest path $SDP(sj\ \ {}^{p*}_{s}, v$
)24. end if
25. if tj is not equal to $v_t^{p*}$
26. $SDP(v_t^{p*}, tj)$ ← Compute shortest path $SDP(v_t^{p*}, tj)$
27. end if
28. Return $p* = SDP(sj, v_s^{p*})$ Ⓢ$p*$ Ⓢ$SDP(v_t^{p*}, tj)$.
```

C. Mathematical Model

Here, Denoting the n^{th} node on the path p_i as v_n, our System use $P\ T\ (p_i, p_j)$ to indicate whether path p_j is a k^{th} PPattern for path p_i.

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, e.g. s = a or b = t. The unshared part 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[1]:

$$|SDP\,(s^j, t^j)| = |SDP\,(s^j, a)| + |SDP\,(a, b)| + |SDP\,(b, t^j)| \quad \text{.............................(1)}$$

For determining whether to store a new query into cache or not cache replacement policy uses the usability factor[1]. Usability is the number f times path used in the past and it is denoted by

$$\mu(p_{i,j}) = \sum_{k=1}^{n} W_{vk} \times count(v_k). \qquad \text{............(2)}$$

where,

$\mu(p_{i,j})$ is the usability of path $p_{i,j}$.

v_k is the node on path $p_{i,j}$.

W_{vk} is the weight of node v_k.

$count(v_k)$ is the number of times node v_k queried in past .

D. HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

1) Processor - Intel i5 core
2) Speed - 1.1 GHz
3) RAM - 2GB
4) Hard Disk - 40 GB
5) Key Board - Standard Windows Keyboard
6) Mouse - Two or Three Button Mouse
7) Monitor - SVGA
8) Floppy Drive - 44 Mb

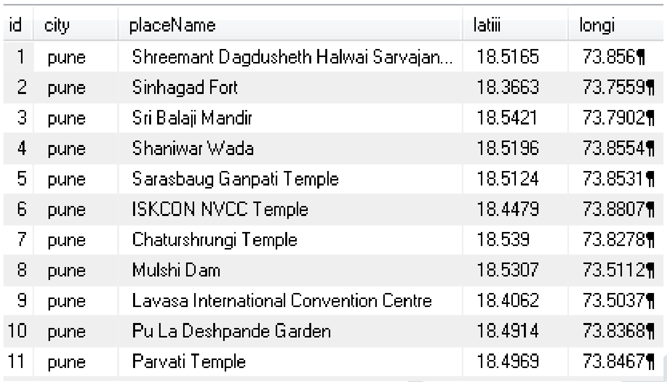
Software Requirements

1) Operating System - XP, Windows7/8/10
2) Coding language - Java, MVC, JSP, HTML, CSS etc
3) Software - JDK1.7
4) Tool - Eclipse Luna
5) Server - Apache Tomcat 8.0
6) Database - MySQL 5.0

E. DATASET

1) City Dataset(Shown in Fig.2)

a) City - Name
b) Place -Name
c) Longitude
d) Latitude
e) 500 records (can be maximized).



id	city	placeName	latiii	longi
1	pune	Shreemant Dagdusheth Halwai Sarvajan...	18.5165	73.856¶
2	pune	Sinhagad Fort	18.3663	73.7559¶
3	pune	Sri Balaji Mandir	18.5421	73.7902¶
4	pune	Shaniwar Wada	18.5196	73.8554¶
5	pune	Sarasbaug Ganpati Temple	18.5124	73.8531¶
6	pune	ISKCON NVCC Temple	18.4479	73.8807¶
7	pune	Chaturshrungi Temple	18.539	73.8278¶
8	pune	Mulshi Dam	18.5307	73.5112¶
9	pune	Lavasa International Convention Centre	18.4062	73.5037¶
10	pune	Pu La Deshpande Garden	18.4914	73.8368¶
11	pune	Parvati Temple	18.4969	73.8467¶

Fig. 2. Dataset

IV. SYSTEM ANALYSIS AND RESULT

In this System, We considered the City as Data Set on which applied the PPattern Algorithms and Shortest path Estimation Algorithms and for Detection of current location use Google API with key. We conduct a comprehensive performance evaluation of the PPC system using the city data set. The data set has many nodes for the query log, we obtain the Points-of-interest (POIs) interaction in P Pattern. Next, we randomly select pairs of nodes from these POIs as the source and destination nodes for path planning queries. cache construction mechanism uses usability value(μ) that measures importance of path that reduces the time required to find out best path. Usability of path $p_{i,j}$ is given above in equation (2) as:

$$\mu(p_{i,j}) = \sum_{k=1}^{n} W_{vk} \times count(v_k) \qquad \text{...[ref.1]}$$

Here ,we have already seen in equation(2) that W_{vk} is weight and $count(v_k)$ is number of occurrences of node. Consider there are two matching paths in cache shown in following table 1. For path $p_{2,4}$ and $p_{1,3}$ consider nodes v_1, v_2, v_3, v_4 with W_{vk} = 0.9, 1, 1, 0.6 and $count(v_k)$ = 2, 1, 2, 1 So, usability of path is computed as $1 \times 1 + 1 \times 2 + 0.6 \times 1 = 3.6$ and $0.9 \times 2 + 1 \times 1 + 1 \times 2 = 4.8$.

Table1:look up two Paths in Cache:

Path	Paths in cache	μ
$P_{2,4}$	$\{P_{v_2,v_3,v_4}\}$	3.6
$P_{1,3}$	$\{P_{v_1,v_2,v_3}\}$	2.3

Now, Performance of the system in terms of hit ratio is given below. Hit ratio is the ratio that user's query matches with shortest path in the cache. Hit ratioδ_{hit} is given as:

$$\delta_{hit} = hits_{cache} / |Q| \times 100\%.$$

Where, $hits_{cache}$ is total number of hits and $|Q$ is the total number of queries given by user's. Following figure shows performance of PPC system in terms of hit ratio. As the size of cache increases hit ratio aslo increases.

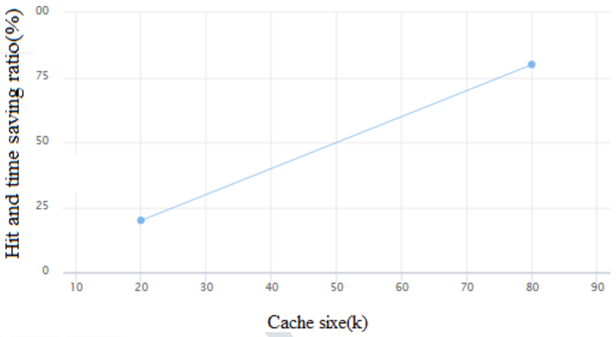


Fig. 3. Result with Cache size(k), k= number of paths

Comparison with other systems: Following figure shows comparison of PPC system with SPC* system in terms of hit ratio. For large number of queries PPC gives more hits as compare to SPC* system. Time complexity of SPC* system $O(n^2)$ is more than PPC system with complexity $O(n)$.

Metric	PPC (%)	SPC* (%)
δhits	18.41	11.47
	46.45	28.13
	31.12	1.19

Fig. 4. Comparison of PPC and SPC* system .....[ref.1]

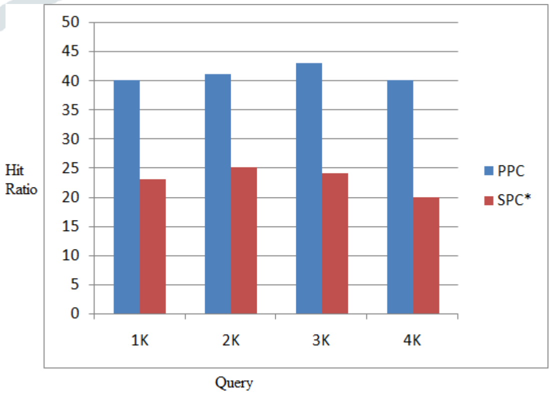


Fig. 5. Comparison of PPC and SPC* system

Following figure shows the comparison of shortest path estimation algorithm in PPC system with the Dijkstra algorithm.

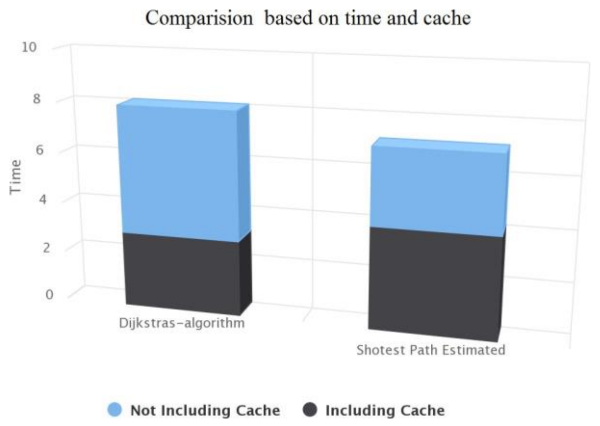


Fig. 6. System Comparison Graph

V. Conclusion

1) Path Planning by Caching (PPC) system is used to answer path query in a timely fashion by reusing historical queried paths from cache.
2) Unlike conventional systems PPC uses partially matched queries. As a result server only needs to compute unmatched part of the query.
3) PPC system significantly reduces overall system workload.
4) Experimentation on road network database shows that this system reduces 32% of the computational latency on an average.
5) PPattern algorithm saves retrieval of path nodes by 31.69% on average.
6) The experimental results shows that cache replacement policy increases the overall cache hit by 25.02%

VI. Future Scope

Miss Ratio Curves(MRC) management technique can be used for cache management instead of cache construction and update algorithm in PPC system as a future work. MRC technique uses cache utility curves that are effective for managing cache allocation. Such curves plot a performance metric as a function of cache size. Miss ratio decreases as cache size increases. MRC can also used to predict future cache performance and for planning and optimization. MRC technique requires less time to locate the shortest path in cache as compare to cache construction an update algorithm in PPC. So, It reduces overall computational latency.

References

[1] Ying Zhang, Member, IEEE, Yu-Ling Hsueh, Member, IEEE,Wang-Chien Lee, Member, IEEE, and Yi-Hao Jhang, “Efficient Cache Supported Path Planning on roads,” *IEEE transaction on knowledge and data engineering*, 2018.

[2] J. R. Thomsen, M. L. Yiu, and C. S. Jensen, “Effective caching of shortest paths for location-based services,” *in ACM International Conference on Management of Data*, 2012.

[3] 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.

[4] H. Mahmud, A. M. Amin, M. E. Ali, and T. Hashem, “Shared Execution of Path Queries on Road Networks,” *Clinical Orthopaedics and Related Research*, vol. abs/1210.6746, 2012.

[5] A. V. Goldberg, H. Kaplan, and R. F. Werneck, “ Reach for A*: Efficient Point-to-Point Shortest Path Algorithms,” *in Workshop on Algorithm Engineering and Experiments*, 2006, pp. 129-143.

[6] R. Gutman, “ Reach-Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks,” *in Workshop on Algorithm Engineering and Experiments*, 2004.

[7] S. Jung and S. Pramanik, “An 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.