International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2020/204020 Volume 7, Issue 5, September – October (2020)

RESEARCH ARTICLE

A Comparative Study on Efficient Path Finding Algorithms for Route Planning in Smart Vehicular Networks

Gurpreet Singh Shahi
School of Computer Science and Engineering, Lovely Professional University, Punjab, India.
gurpreet.17671@lpu.co.in

Ranbir Singh Batth
School of Computer Science and Engineering, Lovely Professional University, Punjab, India.
ranbir.21123@lpu.co.in

Simon Egerton
Department of Computer Science and Information Technology, La Trobe University, Victoria, Australia.
S.Egerton@latrobe.edu.au

Published online: 29 October 2020

Abstract – With the rise in the economy and population of various developing and developed nations leads to an increase in the number of vehicles on the road to manifolds recently. In the smart vehicular networks, there are numerous vehicle transportation problems (VTP) exists. The main or the most common problem is the congestion occurrence due to traffic jams and road accidents, especially in urban areas. So route planning to find the optimal route for a vehicle that is moving from source to destination is an important and challenging task as the route conditions change with the traffic conditions, hence the shortest and optimal route needs to be re-evaluated. This manuscript presents a study and comparison of various pathfinding algorithms which include Dijkstra, A Star, CH, and Floyd Warshall algorithms which are available as optimal route finding algorithms. The simulation of working algorithms with comparison has been performed using SUMO with TRACI and using python coding. The simulation is performed on real maps of urban areas. The parameters used for the comparative study are travel time, distance travelled, and speeds of vehicles.

Index Terms – Smart Vehicular Networks, Guidance Systems, Vehicle Transportation Problem, Routing Algorithms.

1. INTRODUCTION

Smart vehicular networks are non-infrastructure based wireless networks where central administration is not used for communication between vehicles. A unified and basic communication channel is implemented among the vehicles due to the overlapping transmission range of every vehicle in the network. The adaptability of vehicular networks helps in creating ways to a number of applications that can be useful to provide comfort and safety to the passengers [1]. As the concept of a smart city is evolving for the past few years for urban areas which include the collection of information from various sensors and devices as shown in Figure 1. The collected data is then analyzed and based on this some intelligent decisions are taken to improve the various operations of the city [2, 3].

There are various applications of smart cities which include:

- Smart transportation system
- Smart water supply management
- Smart garbage management system
- Smart parking system
- Smart surveillance system
- Smart power system

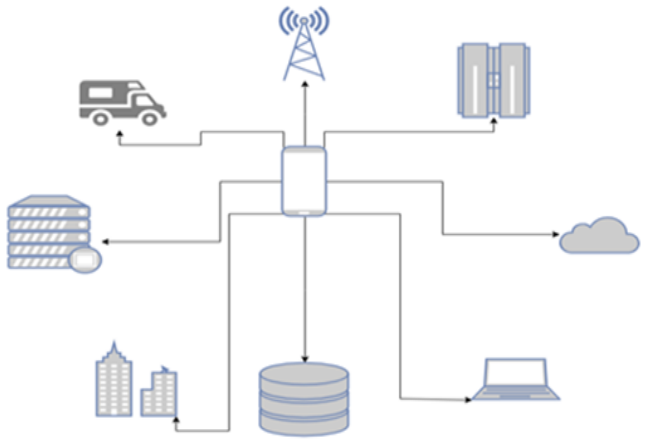

Figure 1 Connected Devices

RESEARCH ARTICLE

To implement these applications various challenges come into the picture, especially for smart transportation systems. As the number of vehicles on urban roads is increasing day by day, so the problem of traffic congestion is also becoming a major problem in urban areas [4]. The average travel time for the vehicles to reach their destination is now increased. Car navigation systems like GPS and GLONASS provide some good solutions in finding the best route for vehicles but these systems are also prone to errors [5]. There may be signal disruption in cities due to buildings, trees, and bad weather. So the real-time information may get delayed. For this reason LPS (Local Positioning Systems) provides better solutions for the quick decision-making process in optimal pathfinding for the vehicles. Now the major challenge is to find the optimal path between two locations in vehicular networks [6, 7]. For this various shortest path algorithms have been proposed like Dijkstra, Astar, Floyd Warshall, etc. But the other challenge is that the road conditions are dynamic as they may change frequently, in this case finding the optimal route is more challenging.

Just like graphs the road network is assumed to be made up of nodes and edges which are connected [8]. The vehicles running on the network are connected and to the roadside units (RSU). The road network is assigned with road id and lane id. The information collected on particular lanes of the road about vehicles defines the congestion density of the vehicles. If congestion is found more, then re-routing decisions have to be made [9, 10]. Here is another problem, In case of congestion on a particular road or lane, all the traffic on the congested lane is re-routed to the same alternate route, then congestion on the alternate route also increases [11]. To avoid this problem a more dynamic and adaptive approach for finding the optimal path of traffic is required. The diagram for the traffic routing algorithm case of congestion is mentioned in Figure 2.

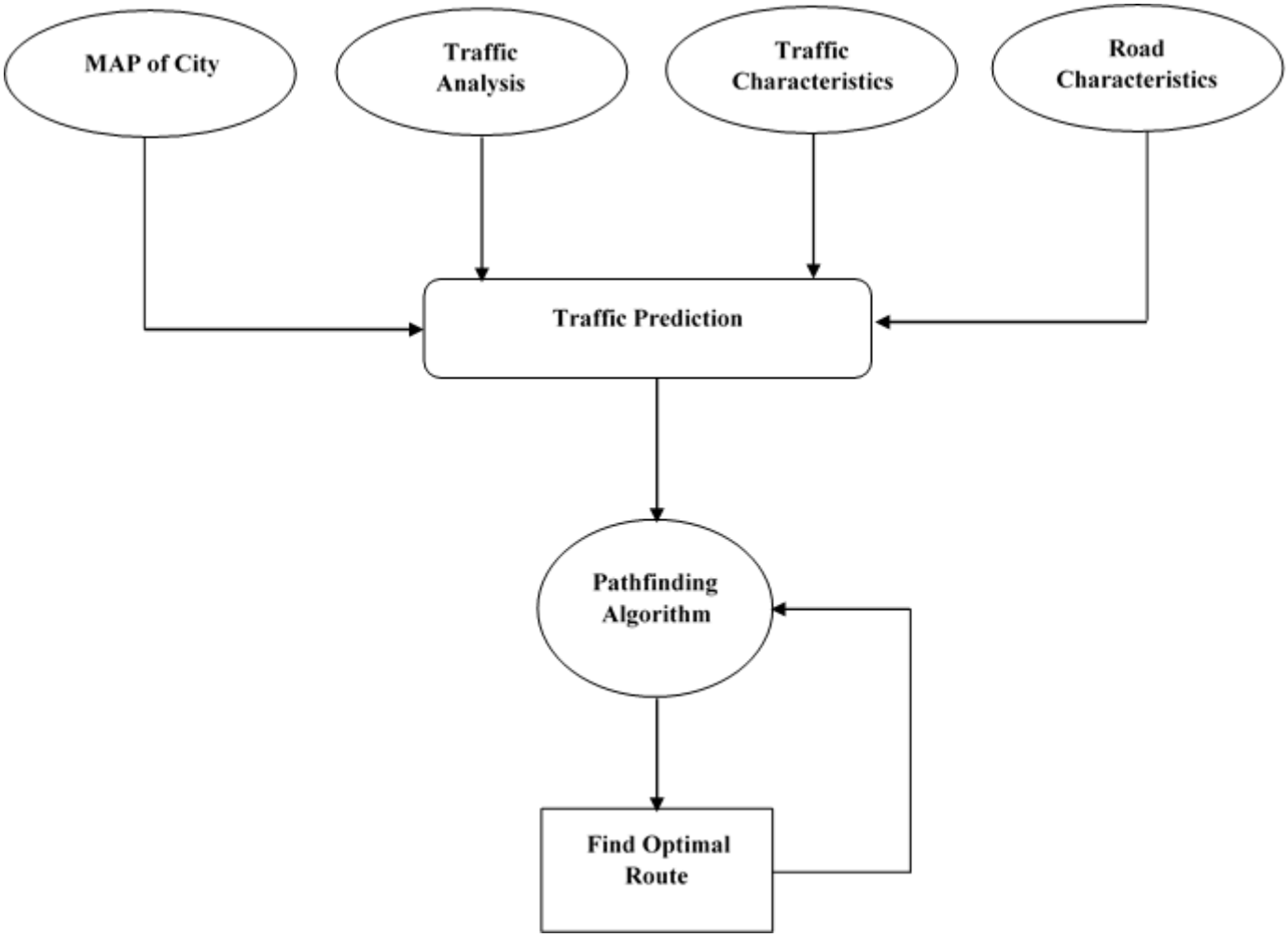

Figure 2 Traffic Routing Algorithm

RESEARCH ARTICLE

To compare the performance of various algorithms used for optimal and shortest path finding, an open-source microscopic road simulator (SUMO) is used. It provides the environment for the simulation of urban mobility traffic systems. To control the traffic scenario TRACI (Traffic control interface) is used with SUMO. The language used to simulate the traffic environment is python. Various algorithms like Dijkstra, Floyd Warshall, and A star, etc are compared and their performance is going to be evaluated based on the average travel time and other parameters. The rest of the paper is organized as follows: Section 2: It describes the works related to various routing techniques and mechanisms that contributed towards the traffic management systems. Section 3: The system architecture consisting of various routing algorithms is explained. Section 4: Simulation results of existing algorithms and performance evaluation is done in this section. Section 6: This section discussed the conclusion of the performance of the various routing algorithms.

2. RELATED WORK

In 2002, I Chabini et al. [12] proposed a dynamic A star algorithm for vehicular networks. To find the optimal solution it uses the cost estimation function to find the scope of the route. M.G.H. Bell et al. [13] in 2012 proposed a hyperstar solution to find a fast and reliable route during congestion. It computes more than one path from the point of congestion. P.Khan et al. [14] in 2014 proposed a modified Floyd Warshall algorithm which uses the directed graph to find the shortest route. In addition to this, there is a confirmation path is available for each traversed path. In 2018, S Hamrioui et al. [15] had researched the area of IoT communications for smart cities to design a smart and self-organized routing (SSR) mechanism. In this research work, the authors try to solve the communication problems in smart cities by analyzing the performance degradation of IoT networks during communications and they proposed a new routing mechanism to improve the network Quality of Services (QoS).

The proposed SSR mechanism works based on the best path selection approach according to the data packets requirements and can reorganize the discovered route during simulation to select the best possible path. That means, designed SSR mechanism by the authors is an adaptive technology that can allow a self-organizing routing process according to the new-fangled circumstances of the IoT network. To validate the routing mechanism, authors compare the SSR mechanism with existing work on the behalf of QoS parameters such as performance in terms of packet delivery rate (PDR), throughput, delay, and overhead packets with the energy consumption. The rate of energy consumption of the SSR mechanism is more regarding smart cities and need to optimize energy consumption and also consider more QoS parameters for the IoT communication in smart cities. Also, the proposed SSR mechanism can be used for simple as well as more complex and smart houses and cities by utilizing the artificial intelligence concept. To improve the communication performance of routing mechanisms for smart cities scenario, K Zahedi et al. [16] in 2019 presented a Connected Junction-based Routing (CJBR) mechanism for VANETs. Authors designed a new routing mechanism to improve the network connectivity by utilizing the CJBR mechanism in the city scenarios and the main aim of researchers is to eliminate the dependency of the routing mechanism on the paths construction's process based on the traffic density inside the road segments. For this purpose, CJBR deploys a selection mechanism for the multi-metric junction which depends on several parameters, and tries to find out the best route using the junction for the data packets transmission. The QoS performance of the CJBR mechanism was improved in terms of the PDR and delay as compared to other existing junction-based protocols at high and low connectivity cases but in case of high mobility, performance degrade.

In 2019, S Haider et al. [17] also researched a robust routing mechanism in VANETs for the broadcasting of warning messages. A Direction Aware Best Forwarder Selection (DABFS) routing mechanism is presented by the authors in this research work to achieve better performance of the network by selecting the best path for data transmission in dynamic nature. DABFS routing mechanism considers the directions as well as the vehicle's relative geographical position to create the best route based on the distance parameter to determine a vehicle's movement direction in the network by utilizing the concept of Hamming distance. This routing mechanism can forward warning messages through a neighbor and best route discovery to the target vehicles. The simulation results of the network demonstrate that the designed DABFS routing mechanism offers improved maximum throughput with minimum loss rate and also the transmission end-to-end delay is far better than other routing mechanisms. But the security concern about the data packets is not considered into account by the researchers in this research.

M Ye et al. [18] in 2019 had researched to design a Mobility Prediction based Routing (MPBR) mechanism in VANETs for data packets transmission. In recent years, the concept of position-based routing protocols in VANET has attracted the most interest due to the type of changeable or dynamic network behavior of vehicular nodes. In this research, the authors presented a new concept for routing that is known as MPBR, and working is based on neighborhood detection, PDR, and path recovery in the network. Authors combine both predictive forwarding strategy and recovery strategy to detect neighbor's vehicular nodes and transfer data packets by utilizing the predicted position and angles of the nodes or vehicles. To validate the effectiveness and feasibility of the MPBR mechanism, authors compare the simulation results

RESEARCH ARTICLE

with existing protocols results in terms of PDR, end-to-end delay and average hops count in urban scenarios respectively.

In 2018, Jhaveri et al.[19] presented a Trusted Routing Scheme (TRS) in which distinct parameters have been used to discover the patterns. The lost packets have been detected through this scheme in which behavior and pattern of the packets were analyzed. Furthermore, three distinct attack patterns have been discovered using the trust model, discovery mechanism, and routing mechanism. The adversaries have been separated at the initial stage to maintain the QoS paradigm. The sensitivity analysis and detection rate have also been optimized using the distrust threshold for better and effective results but the routing overhead during transmission of data packets increases with time and to resolve this problem, an ideal node-based energy consumption model located on game theory was designed by Zhang et al. in 2018 [20].

This document also compared the residual energy and the efficiency under the Nash Equilibrium in a cooperative game and non-cooperative game environment. The results have shown that after nesting, the cluster simultaneously sends the same amount of information and energy efficiency of the optimal gaming model. There are advantages for collaboration for the sensor nodes as the number of energy-saving nodes increases the efficiency raises from 12.870 to 38.796%. This paper also has confirmed the ideal gaming model of energy consumption and collaboration for nodes with better stability in which nodes collaboration has better stability. Authors focus on routing overhead minimization with minimum energy consumption rate but need to elaborate this work with IoT application.

3. SYSTEM ARCHITECTURE

The road network is divided into the edges and vertices. The vehicle moves on-road segments wanting a fast route to reach their destination. The road weight is calculated as the time taken to reach from position A to Position B which is stored in the edge of the road segment. There are two types of pathfinding problems that are available, static and dynamic. The static problems work on fixed graphs while dynamic works on the graph configuration where the number of nodes keeps on changing. So there is a need to update the graph on a subsequent basis. Real-time applications like traffic management systems require dynamic configuration.

To calculate the shortest distance various routing algorithms are there like dynamic routing Dijkstra, Dynamic A Star, and Modified Floyd Warshall. Here we are going to discuss various dynamic algorithms and their functioning.

3.1. Dynamic Dijkstra

The dynamic Dijkstra is used for finding the shortest path problem using a retroactive data structure. In this, the historical sequence of events is maintained which helps in deciding the futuristic events. The priority queue is used for solving the dynamic path problem [21]. Here the graph is made up of G (V, E, and W) where V denotes the nodes or vertex, and E defines the edges' is the weight of edges. Now the weight of the edges varies with increases or decreases with time and with insertion or deletion of nodes. Now there is a need for a retroactive priority queue in Dijkstra which allows performing operations at any point in time. The algorithm is defined in Algorithm 1.

```
1: Function Dijkstra (G, S):
2: Dist[S]:= 0
3: For each vertex V in Graph:
4: if V is not equal source
5:  Distance [V]:= infinity
6: add V to Queue
7: Insert (x, t)
8: while Q is not empty:
9: V: = vertex in Q with min dist [V]
10: remove V from Q
11: Invoke (Del_min (t))
12: for each neighbor U of V:
13: alt:= Dist[v] + Length (V, U)
14: if alt <dist[u]:
15: dist [v]:= alt
 16: else
17: previous[v]:= u
18: return distance
19: End
```

Algorithm 1 Algorithm for Dynamic Dijkstra

3.2. Dynamic A Star

A Star algorithms work well to find an optimal solution in a dynamic environment. It is a best-first algorithm whose is aim is to find the shortest path to the next node with less cost. The parameters for calculating cost is the distance traveled or time travel. The cost estimation of A star is done with function $f(n)= g(n)+ h(n)$ where $g(n)$ is the actual cost of the path from starting node to n node and $h(n)$ is the heuristic function which calculates the estimated cost of the cheapest path. The priority queue is used for the selection of nodes with minimum cost. At every step of this algorithm, the nodes are updated on subsequent times and the vertex with the lowest

RESEARCH ARTICLE

value of f(n) is removed from the searching area. The g(n) and h(n) values are updated accordingly. The A Star algorithm uses LPA (Lifelong planning) method in which it can reuse the previously-stored results for better search [22, 23]. The algorithm for Dynamic A Star is given in Algorithm 2.

1: Preprocessing (G, S):

2: Dist[S]:= 0

3: For each Node or vertex (v) in Graph:

4: Calculate shortest path value f(v)

5: Witness search for each pair {u, v}

6: g (v): Cost of path from starting node

7: h (v) = The heuristic approximation of the value of the node

8: f(v)= g(v) +h(v)

9: Store the previous results on edge nodes

10: Query for each neighbor U of V:

11: path: = Dist[v] + Length (U, V)

12: if path <dist [U]:

13: update dist[u]:= path

14: else

15: previous[v]:= u

16: return distance

17: End

Algorithm 2 Algorithm for Dynamic A Star

3.3. Floyd Warshall

Floyd Warshall algorithm is another good example of dynamic pathfinding. The Floyd Warshall algorithm compares every possible combination of path calculation process between all pairs of vertices. This algorithm is used to find the all pair shortest path problem [24]. This shortest path is computed between all pairs of vertices. The graph is taken as a matrix and then updates the solution matrix by considering all vertices as an intermediate vertex. The algorithm of Floyd Warshall is presented in Algorithm 3.

1: Preprocessing (G, S) as matrix:

2: Dist[S]:= 0

3: For each Node or vertex (v) in Graph:

4: for each edge (s,k)

5: dist[s][k]←w(s, k) (weight of path)

6: for each vertex kdo

7: dist[k][k] ← 0

8: forxfrom 1 to |K|

9: forufrom 1 to |K|

10: forvfrom 1 to |K|

11: ifdist[u][v] >dist[u][k] + dist[k][v]

12: dist[u][v] ← dist[u][k] + dist[u][j]

13: end

Algorithm 3 Algorithm for Floyd –Warshall

3.4. CH Algorithm

Contraction Hierarchies (CH) is a speed-up technique to find the shortest path in a graph efficiently. The CH method exploits the graph for speeding up the traversing system. The CH routing algorithm avoids unimportant vertices and creates shortcut routes for vehicles [25]. The metric used in this algorithm is time travel which is used as the weight for path calculation on the road edges. The approach of CH consists of two phases:

3.4.1. Pre-Processing Phase

Consider that two large cities are connected through a network of roads (highway). The vehicle wants to reach from the source a to destination d. Throughout the highway, there are multiple junctions (edges) that lead to villages and small towns. The CH algorithm pre-processes the information of route calculation of all paths (vertices) that connect the two cities by traversing through all edges and vertices [26]. This information is stored in an additional edge which is called a shortcut edge. Now when the same highway is evaluating for traffic analysis again then pre-processing information that is already stored on the shortcut edge can be used which saves time during the path evaluation query. The working diagram is explained in Figure 3.

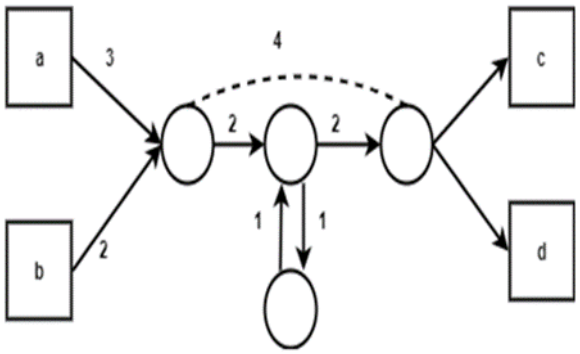

Figure 3 Traversing in CH Algorithm

The CH algorithm depends on the shortcut routes which is found during the pre-processing phase with fewer vertices as defined in the above diagram (shortcut path is represented

RESEARCH ARTICLE

with dotted lines). There are two approaches for traversing the graph.

The first one is bottom-up which does not know the order of the graph and selects the next node when the previous node is visited. Another approach is top-down which computes the node order in the graph before traversing starts. The computation time in the bottom-up approach is less as compared to the top-down approach.

3.4.2. Query Phase

In this phase, a bidirectional search is done starting from the source node to the destination. The original graph is a line [x, y, z, t, u, v]. To calculate the shortest path between x and t (Figure 4). The original distance is dist[x,y] +dis[y,z]+dis[z,t].Now the shortest path is the dist[x,y]+dist[y,t]. The path weights are calculated before choosing the path which includes fewer edges. The weight that is taken here is time travel and this information is stored at the end of each node.

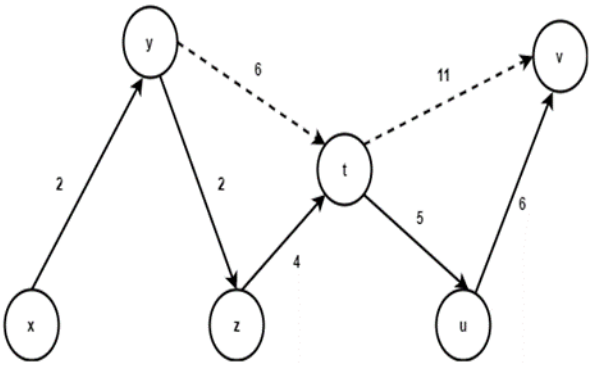

Figure 4 Node Order for Query Phase

The algorithm for CH is depicted in Algorithm 4.

1: Preprocessing (S, D):

2: Dist[S]:= 0

3: For each Node or vertex (x) in Graph:

4: Calculate shortest path value f(v)

5: Witness search for each pair {x, y}

6: s (x): shortcut path value from starting node

7: h (x) = The heuristic of the value of the node

8: v: = vertex in Q with min dist[n]

9: Pre-compute and store information on edge node

10: Query for each neighbor x of y for shortcut path

11: path: = Dist[x] + Length (X,Y)

12: if path <dist [Y]:

13: update dist[y]:= path

14: else

15: previous[x]:= y

16: return distance

17: End

Algorithm 4 Algorithm for CH

4. SIMULATION AND RESULTS

To simulate the routing protocols the Sumo Simulator is used. The Simulator is an open-source urban mobility simulator. The Python script is used for writing the code for simulation. The open street map Los Angeles is used with an area of 10 square km. The diagram of the simulation of the Los Angeles map is shown in Figure 5. Figure 6 is showing the running status of vehicles in the streets of an urban city. The vehicle parameters are shown in Table 1. This includes the length of vehicles, its emission standards, maximum speed, and minimum gap between the vehicles, acceleration, and deceleration of the vehicles.

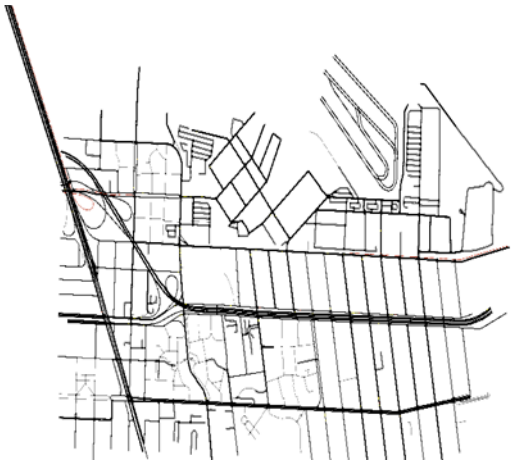

Figure 5 Simulation Los Angeles Map of 10 km

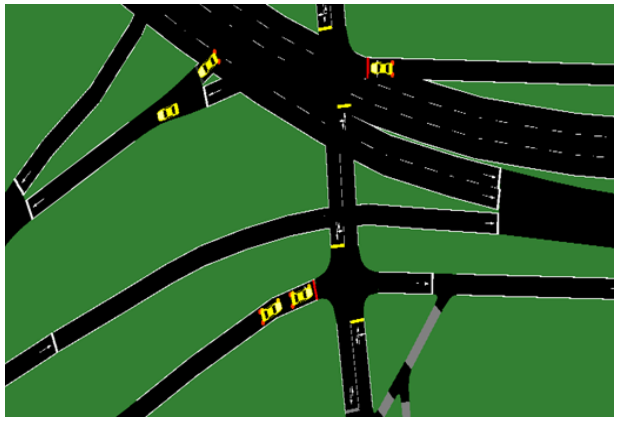

Figure 6 Los Angeles Map Simulation

RESEARCH ARTICLE

Acceleration	Deceleration	Min gap	Length of Vehicle	Emission Class	Max Speed
2.60ms^{-2}	4.50 ms^{-2}	2.50m	5m	HBEFA3/PC_G_ EU4	55.31 Km/hr

Table 1 Parameters of Vehicle Used for Simulation

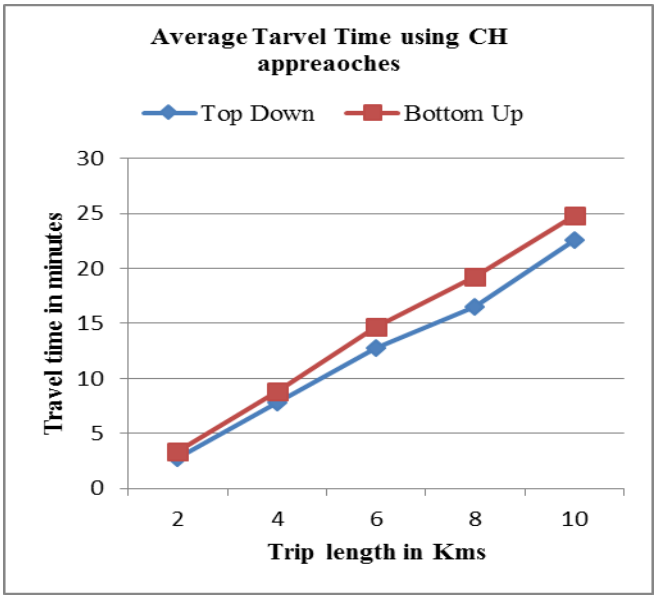

Figure 7 Average Travel Time with CH Approaches

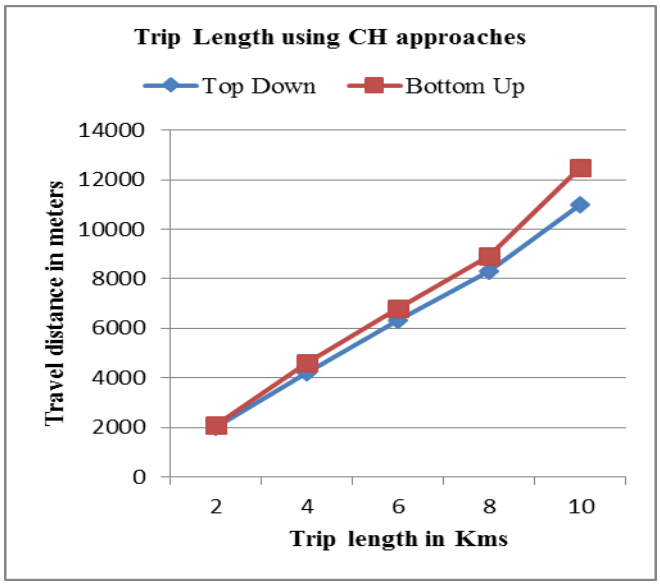

Figure 8 Travel Distance with CH Approaches

The comparative analysis of Dynamic A Star, Dynamic Dijkstra, Floyd Warshall, and CH routing algorithm is done based on the various attributes. The average travel time and an average travel distance of the CH algorithm are calculated based on the Top-down approach and bottom-up approach [See Figure 7 and Figure 8]. It has been found that the Top-down approach is better optimized in terms of travel time and distance. This is due to the calculation of the path. The top-down heuristics computes the whole node order of the graph before the contraction of the graph. This yields better results. The bottom-up heuristic does not compute the node order in advance, it selects the next node for contraction during traversing.

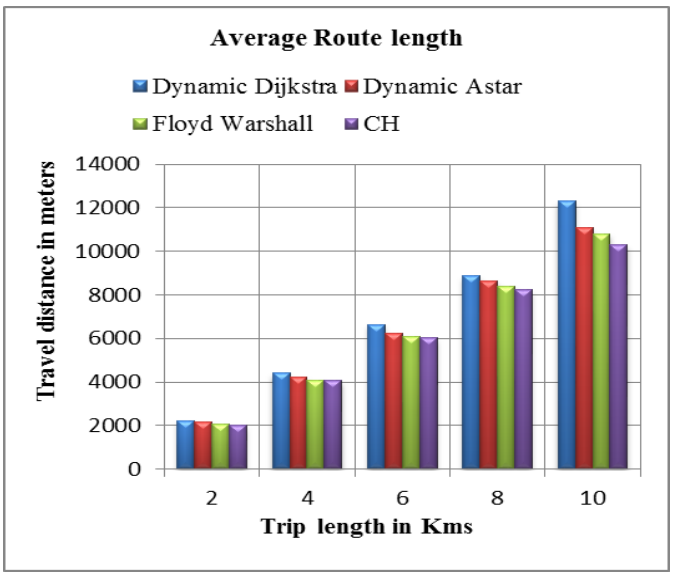

Figure 9 Average Route Length Comparison of Algorithms

The various routing algorithms are compared based on the parameters like route length, average travel time, average waiting time, and average speed of vehicles. Figure 9 shows the average route length covered by vehicles using various routing algorithms. The comparison of the vehicles for a distance of two to ten kilometers is considered. There is no significant difference in route covered by using routing

RESEARCH ARTICLE

algorithms when the distance is small. As the travel distance is increased, it has been found that the CH algorithm covered less distance (route length) as compared to other algorithms. The route length becomes an important parameter especially in case of congestion when rerouting is needed for the vehicles. Figure 10 describes the average travel time comparison of vehicles. The travel time of the CH algorithm is less as compared to other algorithms. Figure 11 represents the average waiting time of different algorithms used for simulation. The A Star algorithm is found to be effective, as waiting time is less during the travel of the vehicles as compared to other algorithms.

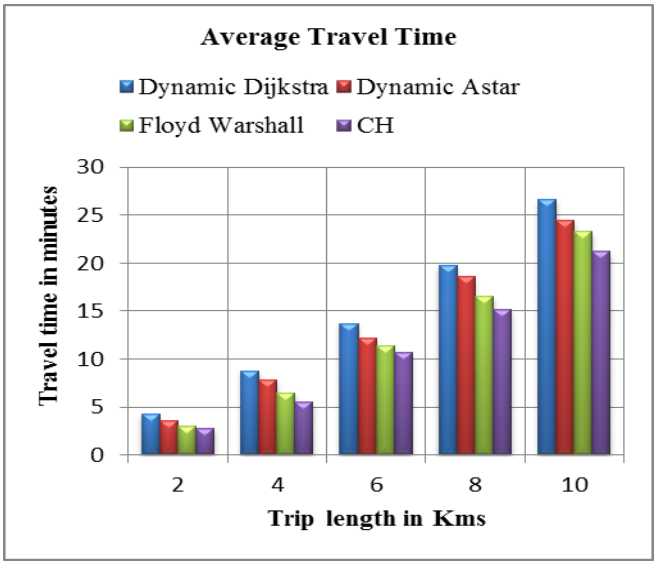

Figure 10 Average Travel Time Comparison of Algorithms

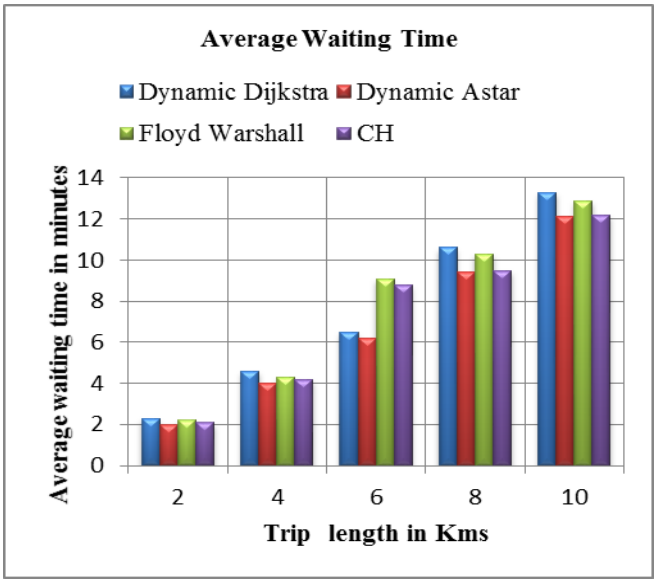

Figure 11 Average Waiting Time Comparison of Algorithms

Figure 12 represents the speed comparison of vehicles using four different routing algorithms. The speed of vehicles using the CH algorithm is found to be good as compared to other algorithms. The various statistics of routing algorithms according to trip length, route length, average waiting time, average travel time, and average travel speed of vehicles is described in Table 2, Table 3, and Table 4 for the four algorithms.

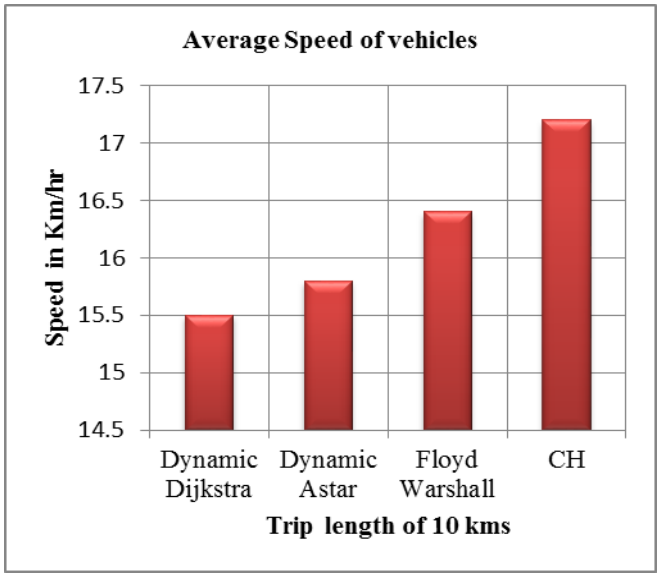

Figure 12 Average Speed Comparison of Algorithms

Parameters / Algorithms	Route Length in Meters	Average Waiting Time in Minutes	Average Travel Time in Minutes
Dynamic Dijkstra	2240	2.3	4.3
Dynamic A Star	2182	2	3.6
Floyd Warshall	2050	2.2	3
CH	2012	2.1	2.8

Table 2 Statistics for Trip Length for 2 Kms

RESEARCH ARTICLE

Parameters / Algorithms	Route Length in Meters	Average Waiting Time in Minutes	Average Travel Time in Minutes
Dynamic Dijkstra	12300	13.3	26.6
Dynamic A-star	11076	12.1	24.5
Floyd Warshall	10800	12.9	23.3
CH	10300	12.2	21.2

Table 3 Statistics for Trip Length for 10 Kms

Algorithm	Dynamic Dijkstra	Dynamic A Star	Floyd Warshall	CH
Average Speed(km/hr)	15.5	15.8	16.4	17.3

Table 4 Statistics for Average Speed for 10 Kms

5. CONCLUSION

In this manuscript, the evaluation of the four routing algorithms has been done using SUMO simulation. The performance of algorithms is evaluated based on the average route length covered, the average waiting time of vehicles, average speed of vehicles, and average travel time of vehicles using an open street map of Los Angeles of 10 KM area. The python script is used to implement the algorithms. The order of implementation of the routing algorithm is defined below:

CH >FloydWarshall> Dynamic Astar>Dynamic Dijkstra.

Contraction hierarchies' results are found to be good in finding the shortest path, it reduces travel time and the average speed of the vehicle is also improved. CH algorithm is analyzed with top-down and bottom-up approaches. The performance of the top-down approach is found good as compared to the bottom-up approach in terms of finding the optimal pathfinding solution. This is due to the pre-computation of the path in advance in the top-down approach. The average waiting time of CH is found to be little more than other algorithms. In the future, the work needs to be done to reduce the computational time of the algorithms so that quick decisions can be made for the traffic management system.

REFERENCES

[1] Rafsanjani, Marjan Kuchaki, Hamideh Fatemidokht, Valentina Emilia Balas, and Ranbir Singh Batth. "An Anomaly Detection System Based on Clustering and Fuzzy Set Theory in VANETs." In International Workshop Soft Computing Applications, pp. 399-407. Springer, Cham, 2018.

[2] Conti, M., Boldrini, C., Kanhere, S. S., Mingozzi, E., Pagani, E., Ruiz, P. M., & Younis, M. (2015). From MANET to people-centric networking: Milestones and open research challenges. Computer Communications, 71, 1-21.

[3] M. Zhou, "Internet of Things: Recent advances and applications", Proc eedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2013.

[4] L. Atzori, A. Iera and G. Morabito, "Introduction," The Internet of Thi ngs: A survey, Computer Networks, vol. 54, no. 15, pp. 2787- 2788, 2010.

[5] G. Owojaiye and Y. Sun, "Focal design issues affecting the deployment of wireless sensor networks for intelligent transport systems," IET Intelligent Transport Systems, vol. 6, no. 4, pp. 421–432, 2012.

[6] S. Wang, S. Djahel, Z. Zhang, and J. Mcmanis, "Next Road Rerouting: A Multiagent System for Mitigating Unexpected Urban Traffic Congestion," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2888–2899, 2016.

[7] S. G. Manyam, S. Rathinam, S. Darbha, D. Casbeer, and P. Chandler, "Routing of two Unmanned Aerial Vehicles with communication constraints," 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014.

[8] N. Akhtar, S. C. Ergen, and O. Ozkasap, "Vehicle Mobility and Communication Channel Models for Realistic and Efficient Highway VANET Simulation," IEEE Transactions on Vehicular Technology, vol. 64, no. 1, pp. 248–262, 2015.

[9] Shahi, G. S., Batth, R. S., & Egerton, S. (2020). MRGM: An Adaptive Mechanism for Congestion Control in Smart Vehicular Network. International Journal of Communication Networks and Information Security, 12(2), 273-280.

[10] A. Nayar, R. S. Batth, D. B. Ha, and G. Sussendran, G. "Opportunistic networks: Present scenario-A mirror review"InternationalJournal of Communication Networks and Information Security," 10 (1), pp. 223-241, 2018.

[11] R. S. Batth, M. Gupta, K. S. Mann, S. Verma, and A. Malhotra, "Comparative Study of TDMA-Based MAC Protocols in VANET: A Mirror Review," Advances in Intelligent Systems and Computing International Conference on Innovative Computing and Communications, pp. 107–123, 2019.

[12] Chabini, Ismail, and Shan Lan. "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks." Intelligent Transportation Systems, IEEE Transactions on 3.1 (2002): 60-74

[13] Bell, M. G., Trozzi, V., Hosseinloo, S. H., Gentile, G., & Fonzone, A. (2012). Time-dependent Hyperstar algorithm for robust vehicle navigation. Transportation Research Part A: Policy and Practice, 46(5), 790-800.

[14] P.Khan, G.Konar, &N.Chakraborty (2014). Modification of Floyd-Warshall's algorithm for Shortest Path routing in wireless sensor networks.2014 Annual IEEE India Conference (INDICON). doi:10.1109/indicon.2014.7030504

[15] Hamrioui, Sofiane, Camil Adam Mohamed Hamrioui, Jaime Lioret, and Pascal Lorenz. "Smart and self-organised routing algorithm for efficient

RESEARCH ARTICLE

IoT communications in smart cities." IET Wireless Sensor Systems 8, no. 6 (2018): 305-312.
[16] Zahedi, Khalid, Yasser Zahedi, and Abdul Samad Ismail. "CJBR: connected junction-based routing protocol for city scenarios of VANETs." Telecommunication Systems 72, no. 4 (2019): 567-578.
[17] Haider, Shahab, Ghulam Abbas, ZiaulHaq Abbas, and Thar Baker. "DABFS: A robust routing protocol for warning messages dissemination in VANETs." Computer Communications 147 (2019): 21-34.
[18] Ye, Mao, Lin Guan, and Mohammed Quddus. "MPBRP-Mobility Prediction Based Routing Protocol in VANETs." In CommNet, pp. 1-7. 2019.
[19] R. H. Jhaveri, N. M. Patel, Y. Zhong, & A. K. Sangaiah, "Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial IoT," IEEE Access, vol. 6, pp. 20085-20103, 2018.
[20] J. Zhang, J. Yin, T. Xu, Z. Gao, , H. Qi, & H. Yin, "The optimal game model of energy consumption for nodes cooperation in WSN. Journal of Ambient Intelligence and Humanized Computing, pp. 1-11, 2018.
[21] Sunita, &D.Garg, (2018). DynamizingDijkstra: "A solution to dynamic shortest path problem through retroactive priority queue". Journal of King Saud University - Computer and Information Sciences. doi:10.1016/j.jksuci.2018.03.003.
[22] Huang, Bo, Q.Wu, and F. B. Zhan. "A shortest path algorithm with novel heuristics for dynamic transportation networks." International Journal of Geographical Information Science 21.6 (2007): 625-644.
[23] Koenig, Sven, Maxim Likhachev, and David Furcy. "Lifelong planningA*."Artificial Intelligence 155.1 (2004): 93-146.
[24] Ramadiani, D.Bukhori, Azainil, &N.Dengen, (2018). Floyd-warshall algorithm to determine the shortest path based on android. IOP Conference Series: Earth and Environmental Science,144, 012019. doi:10.1088/1755-1315/144/1/012019.
[25] R. Singh and K. S. Mann, "Improved TDMA Protocol for Channel Sensing in Vehicular Ad Hoc Network Using Time Lay," Proceedings of 2nd International Conference on Communication, Computing and Networking Lecture Notes in Networks and Systems, pp. 303–311, 2018.
[26] Fu, Liping, D. Sun, and Laurence R. Rilett. "Heuristic shortest path algorithms for transportation applications: state of the art." Computers & Operations Research 33.11 (2006): 3324-3343.

Authors

Mr. Gurpreet Singh Shahi is a Phd. Research Scholar in the in the School of Computer Science and Engineering at Lovely Professional University, Punjab, India. He has completed B-Tech in Computer Science and Engineering from Beant College of Engineering and Technology, Gurdaspur, Punjab and M-Tech in Computer Science and Engineering from Lovely Professional University, Punjab, India. His research interests are vehicular ad-hoc networks, wireless sensor networks, smart transportation systems and machine learning. He has number of research papers to his credit. He has 7 years of teaching experience as assistant Professor Lovely Professional University, Punjab, India.

Dr. Ranbir Singh Batth is working as an Associate Professor in the School of Computer Science and Engineering and he also serves as a coordinator for International relations, at Lovely Professional University, Punjab, India. He has received his Ph.D. from IKG Punjab Technical University, Kapurthala, Punjab, India in 2018 and the Master degree in Computer Engineering from Punjabi University, Patiala. His research interests include Wireless Sensor Networks, Cloud Computing, Network Security, Ad Hoc Networks, IoT, Machine Learning, Deep Learning, Wireless Communications and Mobile computing. He also serves as an editorial member, guest editor, and reviewer for various reputed International journals. He has been the organizing chair, session chair and advisory member for various reputed International conferences. He is an active member of ACM and IEEE computer Society.

Dr. S Egerton is currently Associate Professor and Deputy Head in the Department of Computer Science and Information Technology at La Trobe University, Australia. He is cofounder of the Creative Science Foundation and leads the Disruptive Technology Research Group at La Trobe specializing in applied Internet of Things research. He obtained his doctorate in the field of robotics from the School of Computer Science and Electronic Engineering at the University of Essex in the UK. His research interests cover low power wide area networks, machine learning, artificial intelligence and applications to smart cities, farms and improving rural health and well-being.