

AMBULANCE OPTIMAL ROUTING PATH DETECTION USING FOLIUM AND POLYGONAL PATH FINDING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

by

V. JANAKI RAMAYYA (39130517)

Y. HRUTHIK CHARAN (39130518)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICAITON ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of V.Janaki Ramayya (39130517) and Y.Hruthik Charan (39130518) and who carried out the project entitled "AMBULANCE OPTIMAL ROUTING PATH DETECTION USING FOLIUM AND POLYGONAL PATH FINDING" under our supervision from November 2022 to April 2023.

Internal Guide

MS.MARY SAJIN SANJU I, M.TECH.(Ph.D)

ASSOCIATE PROFESSOR, Department of ECE

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 19-04-2023

Internal Examiner

External Examiner

DECLARATION

We V. Janaki Ramayya (39130517) and Y. Hruthik Charan (39130518) here by declare that the Project Report entitled "**AMBULANCE OPTIMAL ROUTING PATH DETECTION USING FOLIUM AND POLYGONAL PATH FINDING**" done by us under the guidance of Ms.MARY SAJIN SANJU I, M.TECH.(Ph.D) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 19-04-2023

PLACE: CHENNAI.
CANDIDATE(S)

SIGNATURE OF THE

1. V. Janaki Ramayya
2. Y. Hruthik Charan

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Professor & Dean, School of Electrical and Electronics** and **Dr. T. RAVI, Ph.D., Professor & Head, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our **Project Guide Ms.MARY SAJIN SANJU I, M.TECH.(Ph.D)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Disaster management generally includes the post-disaster stage, which consists of the actions taken in response to the disaster damages. These actions include the employment of emergency plans and assigned resources to (i) rescue affected people immediately, (ii) deliver personnel, medical care and equipment to the disaster area, and (iii) aid to prevent the infrastructural and environmental losses. Researchers pointed out the importance of equity and fairness in humanitarian relief services: managing the operations of ambulances in the immediate aftermath of a disaster must be done impartially and efficiently to rescue affected people with different priority in accordance with the restrictions. Our research aim is to find the best ambulance tours to transport the patients during a disaster in relief operations while considering fairness and equity to deliver services to patients in balance. The problem is formulated as a new variant of the team orienteering problem with hierarchical objectives to address also the efficiency issue. Due to the limitation of solving the proposed model using a general-purpose solver, we propose a new hybrid algorithm based on a machine learning and polygon path finding search. Based on a new set of realistic benchmark instances, our quantitative analysis proves that our algorithm is capable to largely reduce the solution running time especially when the complexity of the problem increases. Further, a comparison between the fair solution and the system optimum solution is also provided.

In order to ensure the ambulance arrival to patient is within the targeted time, ambulance availability must be ensured and the time taken to arrive can be controlled. Optimal route that able to provide shortest distance must be determined. Thus, an interface is developed to help the ambulance providers in sending the ambulance to a specified ambulance station and emergency site. The interface for the ambulance routing is designed to give shortest path and shortest distance (in km). The results obtained were programmed using software and the A* algorithm utilized as the engine to determine the shortest distance for the ambulance in the study area.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	11
3	METHODOLOGY	16
	3.1 Existing System	16
	3.2 Proposed System	17
	3.3 Architecture Platform of Proposed Algorithm	18
	3.4 Difference Between Existing and Proposed System	19
	3.5 Advantages of Proposed System	19
	3.6 Risk Analysis	20
	3.7 Software Requirements Specification Document	20
	3.8 Functionalities	21
4	MODULES AND ALGORITHMS USED	23
	4.1 Modules	23
	4.2 Dataset	24
	4.3 Algorithm and Techniques	24
	4.4 Explanation	26
	4.5 Heuristics	28
	4.6 Python Library Used	32
	4.7 Proposed Algorithm as Shortest Path Algorithm	35
	4.8 Pseudocode	35

5	RESULTS AND DISCUSSION	36
	5.1 Results	36
	5.2 Discussion	37
6	CONCLUSION AND FUTURE WORK	39
	6.1 Conclusion	39
	6.2 Future work	39
	6.3 Research Issues	40
	6.4 Implementation Issues	41
	REFERENCES	42
	APPENDIX	44
	A. SOURCE CODE	44
	B. SCREENSHOTS	46
	C. RESEARCH PAPER	48

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Requests Classification and changes in Patient Condition	2
1.2	Statistics about location of hospitals, population	3
3.1	Architecture Platform of Proposed Algorithm	19
4.1	Shortest distance between the nodes	31
4.2	Creating a map using python library folium	33
5.1	Input Result	37
5.2	Plotting the Shortest Path Result	38

CHAPTER 1

INTRODUCTION

In today's world, fortunately, it is possible to cure many diseases and injuries with the development of medical technology. However, obtaining timely aid is crucial for the patient, especially in accidents. Under such circumstances, an ambulance reaching the event scene in a timely manner has a significant economic and social influence. There has been news reporting the death of patients due to the untimely rescue of ambulances. Issues such as the untimely departure of the ambulance, being unfamiliar with the routes and delays caused by congested traffic are the main reasons resulting in this.

For example, statistics indicate that the chance of survival decreases by 24% for every additional minute of delay in the treatment of cardiac arrest patients. Hence, seeking an effective path, especially in congested traffic, for ambulances is one of the notably demanding research areas because of its vital and direct impact on human lives. With the increase of city size, emergency rescue system is playing an important role in the safety of human life and social security. The Emergency Medical Services (EMS) ambulance is designed to give medical care or treatments to patient at the emergency sites and/or directly send the patients to hospital where intensive care by doctors can be given.

The EMS ambulance provided is equipped with basic life support equipment to help whenever an emergency occurs. There are various technologies and tools used in ambulance optimal routing path detection, such as GPS, real-time traffic data, and machine learning algorithms. By leveraging these technologies, EMS providers can quickly and accurately identify the best route to take, reducing response times and improving overall patient care. This not only helps to save lives but also reduces the risk of complications and improves patient outcomes.

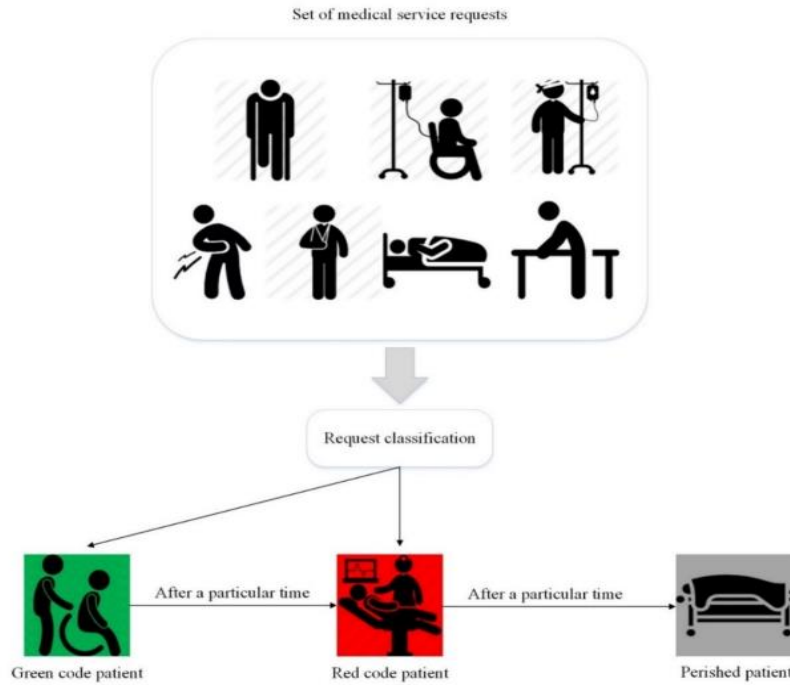


Fig.1.1: Requests Classification and Changes in Patient Condition

Following protocols and guidelines, EMS provides emergency care and ambulance to the patients and involves with patients life or death. Moreover, EMS also provides non-emergency standby duties with minimal charges, for instance, the festivals, sports, motor racing, national and international conference, duties during aircraft emergency landings or crashes and transfer of patients from a hospitals to other hospital.

In order, to ensure the ambulance can arrive to patient within the target time, ambulance availability must be ensured and the time taken to arrive must be controlled. Thus, in order to help improving ambulance system, efficient ambulances' management and system is required to increase the standard of EMS.

In many instances vehicles and hence users' transit time within the network are negatively affected because of delays occasioned by traffic lock jams. In particular, in day to day life, an Emergency vehicle or a Fire vehicle may get stuck aggravating the precarious situation of individuals on board for treatment or early intervention at scenes

where lives and properties may be saved. There exist various factors affecting the travel times of emergency vehicle (emergency vehicle).

The inevitable factors are the traffic condition at different times of the day, random congestions (e.g. due to accidents). Traffic congestion is one of the most fundamental problems in large cities like Lagos. There is a positive connection between the delay of emergency vehicles and ratio of fatal or serious preventable incidences. These untoward losses can be minimized by deploying intelligent decision system and shortest path algorithms to optimize ambulance travel time to and from accidents scene to the nearest hospitals.

At this point, as a sub-field of transportation in health care, seeking an effective emergency medical services (EMS) management plan has been widely studied. Since those plans have a direct effect on human lives, many researchers are dedicated to optimizing them by satisfying all the constraints at the best point, such as minimizing responsiveness time, maximizing availability, and so on.

Such a large number of patients in India lose their lives because the ambulance does not arrive on time or delay in the treatment. Also, in view of the increasing technology and other factors, it is necessary to be aware of such technologies to reduce this problem.

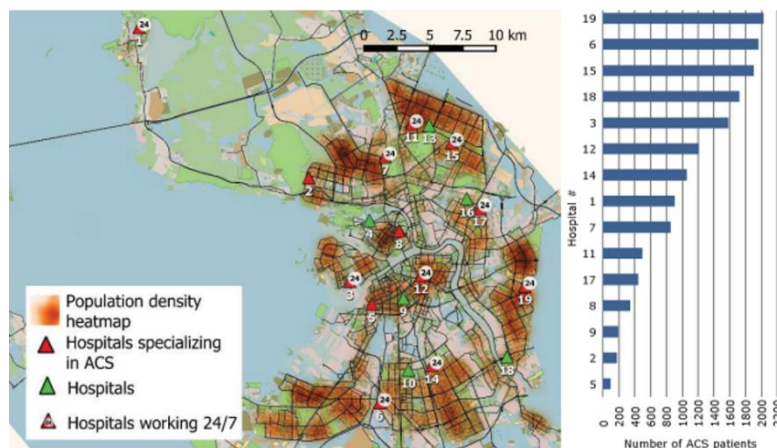


Fig.1.2: Statistics about location of hospitals, population density in the hospitals released during a survey in different hospitals of St. Petersburg in 2014.

A survey conducted in St Petersburg found that the rest of the countries are aware of the technology to find out where the ambulance is and alert the patient to get the crisis vehicle at the ideal time. For this, algorithms like weighted graphs, Dijkstra's algorithm, ant colony algorithm, dynamic shortest path have been used in that system. These algorithms are reduced the traveling time by 20%. Therefore, the chances of patient saving increased. In this system only 2 smartphones will have for conversation between patient and ambulance and it is fully automated.

As we have found in India, if there is a accident at any spot, we are called complementary number or crisis number. After informing where the accident took place on that call, the ambulance is then dispatched from the control room at that particular place, and then the control room gives the details of the nearest hospital and then informs to the hospital. With this proposed system, the ambulance can be transported without delay from the accident site to the hospital we can easily find the shortest route between the accident site and the hospital reduce wasting time. The user no longer need to call and check the availability of the ambulance.

Although these plans seem reasonable and sufficient, they are not adapted to the microcosmic level. Factors such as the complex road situation, traffic jams during rush hour, etc., will complicate ambulance driving in the city, thereby influencing travel time. Hence, in addition to pre-determining a global route, a specific path plan corresponding to these dynamics is also expected.

Analyzing the movement pattern has always been of a keen area of interest, may it be automobile, humans, or any other moving object. These movement patterns can help analysts in making a decision related to the behavior patterns of an object. For example, the idea of geo-marketing can be evolved if the pattern of the people who are shopping is observed.

The disaster and accidents can occur naturally, or because of human consequences. These disasters or accidents, on a larger scale, can affect both existing infrastructures and people. Effective logistical operations are difficult to carry out in such circumstances and thus effective medical assistance cannot be delivered on time,

which in turn causes further losses. Therefore, rescue and assist operations must be well organized after a disaster or accidents and must be carried out efficiently.

In an emergency situation, people who have been wounded need to be taken to hospitals first. In order to carry out this mission effectively, we need to know the location and amount of people injured, the amount and capability of ambulances as all these values are the input to the problem solution. We have to establish an emergency schedule after collecting these inputs so that accidents can be moved to hospitals in a short time. The aforementioned issue relates to the Capacitated ambulance Routing Problem (CARP) which is a form of Ambulance Routing Problem (ARP). Victims and their requests are in ARP.

The goal is to reduce the total distance by an ambulance from the ambulance station location to accident location and accident location to hospital for transfer of victims according to emergency occurred. The ambulance is loaded from the accident place with victims and the victims are handed out to the hospital. If the victims are transported are running out the ambulance will return to the nearby ambulance station. CARP proposes different types of solutions: exact solutions, heuristic, or meta-heuristic approaches. Some researchers are proposing approaches for finding the exact solution.

The history of shortest path algorithm can be traced as early as 1968 in which the A*algorithm was established by Hart et al., then, this algorithm is further discussed and extended by Nilsson, Pohl, Pearl and Patel. In 2006, study by Chen et al. proved that A*algorithm faster and better than Dijkstra's algorithm because A*algorithm is an efficient algorithm to solve the shortest path. A*algorithm able to calculate the shortest path as Dijkstra's algorithm does. Lester stated that A*algorithm essentially the same as Dijkstra's algorithm, distinctive by a heuristic that utilized in A*algorithm. Heuristic is defined as finding optimal solution. Heuristic is very fast, thus it can be easily adapted to various objectives such as minimizing total distance with a few changes in the constraint model. Therefore, the A*algorithm able to compute faster than Dijkstra's algorithm

As mentioned by Shukla et al., before the entire computing algorithm can be applied, it was earlier shown that A*algorithm generates good result for the problem of path planning. It was also corroborated by Liang, the A*algorithm can achieve better running time than other algorithm in the road network application. As reported by Taeg-Keun, A*algorithm is a kind of informed search and widely used in finding an optimal route such as travel distance, traffic condition and etc. This is because the location of starting and ending point are known.

Usually, the A*algorithm is used in finding optimal path through a graph and it is central to many problems that includes route planning for a mobile robot. This model then improved by Kala et al, who reported that the problem of robotic path planning can be solved using A*algorithm. Meanwhile, according to Khantanapoka and Chinnasarn, A*algorithm can be used in increasing and searching for shortest distance from starting nodes to end nodes where the end nodes is optimal way.

1.1 PROBLEM STATEMENT

The model involves road usually that is only a one way track. Ambulances in these areas are facing many constraints such as traffic light, toll, and roundabout. If the ambulance has taken the wrong road, they will results to late arrival at the emergency sites. Therefore, this project focuses on developing an ambulance road network for the area under study specifically. We aim to find a suitable mathematical model for estimating ambulance travelling distance. Thus, A*Algorithm model proposed by this study will be used to provide the route with the shortest total distance in our study area.

1.2 OBJECTIVES

Therefore, two objectives are proposed in this paper, which are (i) to build an interface using the software for finding shortest distance of the ambulances, and (ii) to find the minimum ambulance travelling distance by using (i).

1.3 ABOUT THE PROJECT

Over the past few decades, researchers in a wide range of disciplines have been studying optimal path finding problems within a variety of applications. These approaches find an optimal way to traverse a complex medium or network under a diverse set of constraints, and outside influences like weather. Computational geometry and geographical information systems analyze the shortest paths defined by Euclidean distance and other metrics, often with the presence of polygonal obstacles and weighted homogeneous regions. Optimal robot routing problems integrate the system's physical properties and constraints to find fastest or minimum energy consumption paths over various terrain. Naval vessel path-finding and navigation integrate the vessel's hull structure and forces exerted by waves and wind to minimize travel time to a destination. Each aforementioned application adds complexity to the original optimal path-finding problem, while integrating a number of assumptions in each scenario to make the problem more tractable.

These assumptions often simplify the problem such that it is inappropriate for real-life applications. In our research, we relax a number of restrictive assumptions to create an accurate and tractable model suitable for actual implementation. 1 2 We study optimal path finding problems in a direction, location and time dependent environment. Since the objective of a problem depends on the actual application, we do not restrict our analysis to a specific objective function whenever possible. Throughout this dissertation we discuss the problems of minimizing travel time, fuel consumption, and various motions, as well as more general objective functions. The main complexity of the problems arises from the dependence of the constraints and cost function on the location of the mobile agent, the direction it is heading, and time. Additionally, we integrate the system-dynamics to further constrain a feasible path by maximum sharpness of the turn that an agent can make to add another dimension of reality to our model. Our work delivers a more realistic optimal path-finding model while reducing the computational time required to find such a path. This is particularly important since real-time implementation is essential for our applications. In addition, many analytical results

derived here provide insights into the structure of the problem, its objective function, and the optimal solution. These insights provide a closed-form solution to a large subset of problems where additional assumptions are applicable. For such problems, we easily construct the analytical solutions instead of implementing more involved, and often approximate, methods presented in literature.

Natural or human-made disasters can do severe damage to human life. During such catastrophic situations, efficient logistics operations can alleviate the damage severity to a great extent. It is recognized that the majority of victims perish due to receiving untimely medical aid. Therefore, it is crucial to devise a plan that assigns logistic relief vehicles as quickly and efficiently as possible. However, a few studies have addressed the application of Vehicle Routing Problem (VRP) in disaster response.

Analysing the movement pattern has always been of a keen area of interest, may it be automobile, humans, or any other moving object. These movement patterns can help analysts in making a decision related to the behaviour patterns of an object. For example, the idea of geo-marketing can be evolved if the pattern of the people who are shopping is observed. Similarly, different location-aware applications can help urban planning by observing traffic patterns. Approximately 3.5 billion mobile phone users are predicted worldwide in 2020.

A mobile user location is better estimated these days by the techniques that are currently being developed and used by the telecommunication providers. Therefore, mobile user's patterns and activities are sensed by using different mobility data records that are saved by telecommunication companies. The objective of observing mobility data is to see why and when the objects move.

To accomplish these objectives, various data sources are used such as Global System for Mobile communications (GSM) or Global Positioning System (GPS) for analysing and later transforming them into meaningful predicting patterns. The process of predicting patterns is known as Knowledge Discovery (KD), i.e., produced from raw data and converted into meaningful knowledge. A hybrid system for location recognition and prediction which addressed key issues of location-based

1.4 ARP

Finding a global optimal path in networks is a practical problem among communication and transportation. It has attracted much attention from researchers in numerous fields owing to its broad applications. The traditional path finding algorithms usually used the shortest average travel time as selection criteria. But in real networks, traffic condition is affected by many uncertain factors and the travel time is stochastic. Under such circumstances, the network users may face the risk of being late.

For example, when scheduling significant issues (e.g., taking a flight), a network user needs to budget an extra travel time to guarantee a higher reliability of arriving on time. Then, it is very important for researchers to take travel time reliability into consideration for path finding. Thus, the travel time reliability-based path finding problem has attracted much attention by some recently proposed algorithms. However, few of them can guarantee the global optimum of the solution.

The purpose of ARP is to allocate adequate routes to a fleet of ambulances in set $k \in K$ that are in charge of giving aid to patients. Let $P = G \cup R$ signify the set of patients that consists of two patient groups, namely GC patients G and RC patients R . GC patients receive first aid in the field. In contrast, RC patients are required to be directly transferred to available hospitals in set $h \in H$.

Based on the defined sets, $N = G \cup R \cup H$ represents all of the nodes comprising the network. A binary parameter $I_{k h}$ indicates the initial location of each ambulance.

Each ambulance starts the travel from its initial location and finishes it at any hospital. Besides, $A = \{P \times P\} \cup \{P \times H\} \cup \{H \times P\}$ demonstrates the set of connected arcs where $t(i, j)$ is the travel time related to arc $(i, j) \in A$. A service duration s_i is associated with each patient. This parameter denotes the time needed to prepare an RC patient for transportation to a hospital, whereas for GC patients, it determines the time required to provide them with first aid in the field. Since dropping off patients at hospitals also takes a while, a parameter s_h is applied to consider this period.

A capacity caph is considered for each hospital, limiting the number of patients assigned to a hospital. It is presumed that the total hospitals capacity is sufficient to fulfill all of the requests related to RC patients. Furthermore, each ambulance can carry only one RC patient during its direct travel to the nearest hospital that is not fully occupied. On the contrary, in the case of visiting a GC patient, the ambulance can move towards the next patient who may have a green or red status. Three binary variables x_{gi} , x_{ri} , and x_{di} are used to determine the condition of each patient $i \in P$ through the planning period.

A patients' condition can be green, red, or perished. In order to take all RC patients to hospitals regardless of their initial condition, the ambulances must pass over the perished ones. Other employed parameters and variables are described in Table 2. SCT and time threshold concepts are applied to improve the quality of the solutions. SCT is defined to denote the time when an RC patient is dropped off at a hospital, while for a GC patient, it is the time of first aid termination. The first suggested objective function aims to minimize the weighted sum of the latest SCT T_g between all GC patients and the latest SCT T_r between all RC patients.

A planner can use weights w_g and w_r to determine the relative priority of patient groups over each other. Secondly, a time threshold is allocated to each patient group. When a_i that is the visiting time of patient $i \in G$ exceeds the time threshold m_g , the patient turns to an RC patient, which turns x_{gi} to 0 and x_{ri} to 1. Likewise, if a_i exceeds the time threshold m_r for patient $i \in R$, the patient perishes that makes x_{ri} equal to 0 and x_{di} equal to 1. The second objective function is inclined to minimize the number of patients whose condition gets more critical due to not receiving timely medical aid.

CHAPTER 2

LITERATURE REVIEWS

“Toward a distributed strategy for emergency ambulance routing problem”

by Elgarej Mouhcine; Yassine Karouani; Khalifa Mansouri; Youssefi Mohamed in 2018.

The improved method is based on a distributed architecture to compute and find the optimal path for a set of emergency vehicles. Based on the defect of the ambulance route planning, this paper puts forward a new distributed model to find the best way that minimizes the time while taking into consideration the cases of problems that can appear each time such as traffics, speed limit, number of available ambulances, the position of the hospitals.[1]

“Design and Implementation for Ambulance Route Search Based on the Internet of Things” by Xu Wang; Jian Liu in 2011.

In this system, with the help of the RFID tags in the ambulances and wireless sensor nodes on the roads, the dispatch control center in the hospital can collect the real-time traffic conditions where wireless sensor nodes locates. The report messages are sent to the control center by the multi-hop method. Based on the information coming from sensor nodes, the control center forecasts the optical path to provide the fastest route for the ambulances. In this decision process, the control center will ignore some information from the nodes, which have gher traffic jam than the average. Experimental results show that our system can provide the best route to the ambulances. And it also has the more potential applications in wide areas.[2]

“An optimal routing scheme for critical healthcare HTH services — an IOT perspective” by Ashutosh sharma; Rajiv Kumar in 2017.

Two algorithms has been proposed in this paper as CHS and RTCHS. The result of both algorithms shows that the RTCHS algorithm supports for the real-time HTH service very well. The work carried here for the geographic road networks with context to IOT where optimal path for real-time HTH service provided by ambulance in efficient time.[3]

“Disaster management by integration of web services with geospatial data mining” by Elgarej Mouhcine; Yassine Karouani; Khalifa Mansouri; Youssfi Mohamed in 2014.

The researchers create a SMS alert system for government authorities so they can reach on the spot & also we prefer extraction of knowledge rules for further prediction and analysis of accidents by data mining. The system consists of two emergency personnel like police and ambulance facility supporting mobile environment. This can be access on any GPRS enabled mobile device. The same system can be accessed by common user as well as government authorities.[4]

“Software-Defined Content Dissemination Scheme for Internet of Healthcare Vehicles in COVID-Like Scenarios” by Amuleen Gulati; Gagangeet Singh Aujla; Neeraj Kumar; Sahil Garg; Georges Kaddoum in 2021.

Using the optimal path, the interest packet is forwarded among healthcare vehicles in a smart city scenario. The proposed scheme maximizes the probability of finding the requested data while minimizing latency. As the proposed scheme does not involve broadcasting of nterest packets, the problem of network congestion in CCN can be solved to a large extent. In order to facilitate the interoperability of heterogeneous healthcare devices in the network, and to improve the routing flexibility, an SDN

controller is deployed in the IoHV scenario. It maintains the global information of the network in its flow tables. Moreover, to achieve faster response time and lower latency in the IoHV environment, edge devices are utilized instead of central cloud.[5]

“VRescuer: A Virtual Reality Application for Disaster Response Training”

By Vinh T. Nguyen; Kwanghee Jung; Tommy Dang in 2019.

The intelligent ambulance rescuer was introduced as a rescuer/guider to automatically search and find the optimal paths for saving all rescuees. The trainee can interfere in the rescuing process by placing obstacles or adding more rescuees along the ways which cause the rescue agent to re-route the paths. The VRescuer was implemented in Unity3D with an Oculus Rift device, and it was assessed by twenty users to improve the proposed application.[6]

“Energy-Efficient Strategy for Improving Coverage and Rate Using Hybrid Vehicular Networks” by Deepak Saluja; Rohit Singh; Nitin Saluja; Suman Kumar in 2022.

The strategy is to deploy RSUs on either side of the road and use an optimal combination of mmWave and microwave RAT. The essential analysis of such a hybrid system involves the evaluation of parameters based on the analytic model. Hence, this paper analytically obtains the expression for seamless coverage and connectivity. The analysis is also extended to rate and energy efficiency calculations. The analysis is supported by probabilistic models-based simulations that agree closely with computation results. The results claim that the proposed model leads to improved performance in terms of coverage and rate while maintaining the cost and energy efficiency within the limits.[7]

“Identifying effective policies in approximate dynamic programming: Beyond regression” by Matthew S. Maxwell; Shane G.Henderson; Huseyin Topaloglu in 2010.

Dynamic programming formulations may be used to solve for optimal policies in Markov decision processes. Due to computational complexity dynamic programs must often be solved approximately. We consider the case of a tunable approximation architecture used in lieu of computing true value functions. The standard methodology advocates tuning the approximation architecture via sample path information and regression to get a good fit to the true value function. We provide an example which shows that this approach may unnecessarily lead to poorly performing policies and suggest direct search methods to find better performing value function approximations. We illustrate this concept with an application from ambulance redeployment.[8]

“The research of Multi-Layer-Based on Path Planning for generating optimal path” by Matthew S. Maxwell; Shane G. Henderson; Huseyin Topaloglu in 2010.

In this Paper, to provide the Multi-Layer Based on Path Planning System(MLPPS) for generated optimal path. Previously used path planning system, single-layer based on path planning system, local map of the path in a single situation was ensured by the optimal path. However, there is a problem that continuously local map is acquired to update the more than necessary parts of the map. These problems are due to the continuous path update, can maintain the unnecessary paths, so that it is impossible to ensure optimal path planning system. To solve this problem, they propose MLPPS. Unlike single layer path planning system, MLPPS was divided Under Layer Path Planner and Upper Layer Path Planner. Under Layer Path Planner is generated the local path. And Upper Layer Path Planner is transmitted the optimal path to autonomous vehicle. The Arbiter is present between the two layers.[9]

“Navigation Systems Using A* Algorithm”

by Matthew S. Maxwell; Shane G. Henderson; Huseyin Topaloglu in 2021.

For this, the approach taken is to construct a map of a city/town. This map is created using graphs where the nodes and edges represent the landmarks/locations and roads respectively. Traversal of the graph denotes traversal among the city/town. There are multiple graph traversal algorithms such as Best First Search, Bellman-Ford, and Dijkstra; but an optimal algorithm that finds the shortest path with the least time complexity should be chosen and implemented. Hence the A* algorithm is ideal. The benefit of using the A* algorithm is that it provides the optimum path while adhering to the underlying constraints. While A* algorithm is an optimum path-finding algorithm, it works ideally when the graph created has accurate measurements. Scaling is also not an issue as space complexity increases with the size of the graph. A small part of The city of Bengaluru is taken as the map consisting of 83 locations in which 24 locations consist of the possible destinations and an ambulance is considered as the vehicle of traversal to eliminate underlying constraints such as traffic density and variable speed.[10]

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

There exists a wide variety of algorithms that are capable of solving the shortest path problem. However, only several of the most popular conventional shortest path algorithms are as follows: Dijkstra's Algorithm, Floyd-Warshall Algorithm, Bellman-Ford Algorithm, Genetic Algorithm (GA). Researchers continue to develop novel approaches and algorithms that can solve the shortest path problem faster with more variables.

3.1.1 Dijkstra's Algorithm: Dijkstra algorithm is a single-source shortest path algorithm. Here, single-source means that only one source is given, and we have to find the shortest path from the source to all the nodes.

3.1.2 Floyd Warshall Algorithm: The Floyd Warshall Algorithm is for solving all pairs shortest path problems. The problem is to find the shortest distances between every pair of vertices in a given edge-weighted directed Graph.

3.1.3 Bellman-Ford Algorithm: Like other Dynamic Programming Problems, the algorithm calculates the shortest paths in a bottom-up manner. It first calculates the shortest distances which have at most one edge in the path. Then, it calculates the shortest paths with at-most 2 edges, and so on. After the i -th iteration of the outer loop, the shortest paths with at most i edges are calculated.

There can be maximum $|V| - 1$ edges in any simple path, that is why the outer loop runs $|V| - 1$ times. The idea is, assuming that there is no negative weight cycle if we have calculated shortest paths with at most i edges, then an iteration over all edges guarantees to give the shortest path with at-most $(i+1)$ edges.

3.1.4 Genetic Algorithm: Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space.

3.2 PROPOSED SYSTEM

The A* algorithm is applied to determine the shortest path from KKSA to specified incident site. The shortest path obtained is based on incident site and the nearest ambulance location. It works by maintaining a pair of lists, one containing locations on the tile map which N is the path and another one contains the locations. Then, A* algorithm will continue the loop if there is still next steps that could be considered to be available next step and considers its neighbors.

Difficult parts in solving A* algorithm is to create a good heuristic function to determine $h(n)$. Castor stated that heuristic function is used to find an estimate of how long it will take to reach the destination node from start node. A heuristic function differs from an algorithm where heuristic is more of estimation and is not necessarily provably correct.

An algorithm is a set of steps which can be proven to stop on a particular given set of input. Heuristic function in A* algorithm is an arbitrary, where the better heuristic is, the faster and more accurate the solution will be. However, even deciding a good heuristic, the problem still exists. Even with a shortest path example, the heuristic can change, depending on the implementation of the search, and how easy or complicated the heuristic function is going to be. Therefore, the heuristic used in this research is Haversine Formula.

Haversine is the name of trigonometric function. Accordingly in a research, to presuming a spherical Earth with radius, the locations of the two points in spherical coordinates are longitude and latitude. Sinnott mentioned that in navigation, haversine

formulas become an important equation using the great-circle distance between 2 point on a sphere from their longitudes and latitudes. This formula is used to calculate the distance between longitude and latitude in kilometres or miles. This method gives exact results of the distance.

According to Dave and Richard, we take the radius, R equals to 6367 km and also equal to 3965 miles where this is the circumference divided by 2π . For converting decimal degrees to radians, we must multiply the number of degrees by $\pi/180$. To express c in decimal degrees, we multiply the number of radians by $180/\pi$, but be sure to multiply the number of radians by R to get d . In addition to this, according to Long, the haversine distance is more accurate than other formulas in order to find the shortest distance.

3.3 ARCHITECTURE PLATFORM OF PROPOSED ALGORITHM

The proposed system for freshness detection using deep learning combines the use of computer vision, sensor data, and chemical analysis to provide a comprehensive and accurate assessment of food freshness. The system would begin by using computer vision to analyze images of food products, identifying patterns and features that indicate freshness. The algorithm would be trained on a dataset of labeled images, and then used to classify images as fresh or spoiled, or to predict the shelf life of a product. This would provide an initial assessment of freshness based on the visual appearance of the food.

Next, the system would use sensor data to detect freshness. The deep learning algorithm would analyze sensor data, such as temperature and humidity, to detect changes that indicate that the food is spoiled. This would provide an additional layer of information about the freshness of the food. Finally, the system would use chemical analysis to detect freshness. The deep learning algorithm would analyze data from sensors that detect volatile organic compounds (VOCs) emitted by food as they spoil. This would provide a more accurate measurement of freshness, and detect spoiled food before it reaches consumers. The system would combine the information from all three analysis to give a comprehensive assessment of the

freshness of the food, and provide an accurate prediction of when the food is likely to spoil. This would help to improve the efficiency and accuracy of freshness evaluations, and reduce food waste by identifying spoiled products before they reach consumers. Additionally, the proposed system could also have a mobile application that would allow consumers to scan the product and get a freshness score, this would help to promote transparency and improve consumer trust in the food products they purchase.

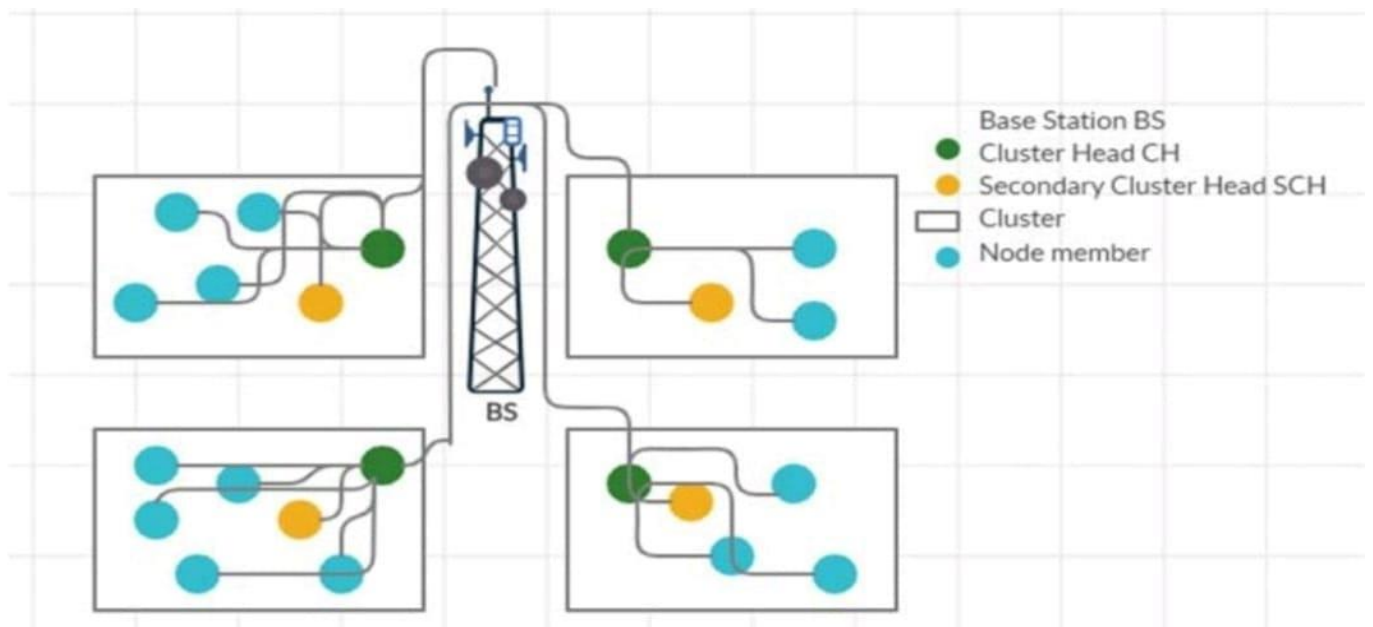


FIG.3.1: Architecture Platform of Proposed Algorithm

3.4 DIFFERENCE BETWEEN EXISTING AND PROPOSED SYSTEM

The existing systems are not very robust in prediction. They have not evolved according to technological advancements and time. The results are also unsatisfactory and may lead to erroneous prediction of optimal path for ambulance leading to unwanted delays in the arrival of patients to the hospital. The proposed system performs better than all previous ones in terms of accuracy and time effectiveness.

3.5 ADVANTAGES OF PROPOSED SYSTEM

- Cost-effective and the data collected is accurate.
- The performance is high because the proposed system deals with accurate data.
- Real-time monitoring is done so that can be operated anywhere and also conveniently.

3.6 RISK ANALYSIS

The risks associated with the project are mainly technological and operational. Technological risks include issues with software, map data, and GPS navigation system. Operational risks include communication issues between ambulance services, hospitals, and the system. In addition, data privacy and security must also be taken into consideration.

To mitigate these risks, the project team must conduct thorough testing and quality assurance procedures. Regular maintenance of the technology and infrastructure must also be performed to ensure smooth operation. Effective communication channels must be established and maintained between all stakeholders. Appropriate security measures must also be put in place to protect sensitive patient information.

3.7 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

The software requirements specification (SRS) document for "Ambulance Optimal Routing Path Detection Using Folium and Polygonal Path Finding" outlines the functional and non-functional requirements of the system. The purpose of this system is to optimize the routing path of ambulances in order to reduce response time and improve emergency medical services. The functional requirements include identifying the location of the ambulance and the destination, identifying the location and traffic status of nearby hospitals, and identifying the optimal route based on

traffic, distance, and hospital availability. The system should also provide real-time updates on the ambulance's location and estimated time of arrival to the destination.

The non-functional requirements include reliability, availability, and security. The system should be reliable and available 24/7 to ensure prompt responses to emergency situations. The system should also be secure to protect sensitive patient information and prevent unauthorized access. In addition, the SRS document also outlines the hardware and software requirements for the system, including the use of GPS and mapping technologies. The document also includes the testing and validation procedures for the system.

Overall, the SRS document serves as a guide for the development team to ensure that the system meets the needs of emergency medical services and provides optimal routing paths for ambulances.

3.8 FUNCTIONALITIES

3.8.1 Input Emergency Location: The system should allow the Emergency Response Team to input the location of the emergency.

3.8.2 Determine Optimal Path: The system should determine the optimal path for the ambulance to reach the emergency location based on factors like traffic, distance, and road conditions.

3.8.3 Provide Navigation Instructions: The system should provide navigation instructions to the Emergency Response Team for them to reach the emergency location.

3.8.4 Update Path: The system should update the optimal path in real-time based on any changes in traffic or road conditions.

3.8.5 Input Hospital Location: The system should allow the Emergency Response Team to input the location of the hospital where the patient needs to be taken.

3.8.6 Determine Optimal Path to Hospital: The system should determine the optimal path for the ambulance to reach the hospital based on factors like traffic, distance, and road conditions.

3.8.7 Provide Navigation Instructions to Hospital: The system should provide navigation instructions to the Emergency Response Team for them to reach the hospital.

3.8.8 Multi-criteria decision-making: The system should be able to consider multiple factors when determining the optimal route, including traffic congestion, road closures, weather conditions, and the urgency of the emergency.

3.8.9 User-friendly interface: The system should have a user-friendly interface that allows ambulance drivers to easily input the destination and receive real-time updates on the optimal route.

3.8.10 Historical data analysis: The system should be able to analyze historical data to identify patterns and optimize future routes based on past experiences.

Folium and Polygonal Path Finding" system aims to improve emergency response times by providing real-time navigation instructions to the Emergency Response Team based on the optimal path determined by the system. The system should also be able to update the path in real-time and provide navigation instructions to the hospital.

CHAPTER-4

MODULES AND ALGORITHMS USED

4.1 MODULES

4.1.1 Setting up of Map

This is a basic step to visualize the geographical data in order to plot & study about the feasibility of the parking spots which is decided by density of the population. Folium is a python library which is used to import map via api call and helps to create a leaflet maps in a separate HTML.

- a) The Required map is imported through the given coordinates, some other coordinates are converted into a format that encodes variety of geographic structures which creates a boundary on the map to create routing.
- b) Using H3, Hexagonal geospatial indexing system is deployed in the given boundary with specified number of hexagons and edge length.
- c) Getting hexagonal coordinates through h3 and plotting the hexagons in the map via folium

4.1.2 Parking Slots

Parking slots are the place where the ambulances are parked over the map to have a easy and fast reachability to the destination of emergency.

- They are represented in JSON data of its coordinates which is converted into a hexagonal format.
- Then it is highlighted using folium in the map to represent the parking slots.

4.1.3 Assigning of Ambulance

- An API is a programming interface which used to respond data when request are sent if we have access to the api.The access is done using a key(a set of characters used to identify the user)
- An API is used to get real time requests from the user and sends the location of the user to our program.
- Isochrone is the a line on a diagram or map connecting points relating to the same time or equal times to reach the destination from a source point.It is used to find the reachability of every ambulance from its parking slot.
- Density.Json is the database for different pathways which is used to connect parking slots to different source points.
- When a request is received from the API, using h3 the program chooses the nearer parking slot which have lesser distance from destination in order to reach fast.

4.2 DATASET

- Initially the data is collected and gathered from both online and realtime sources.RFID tag is installed in the ambulance which will be used as a sensor in case of emergency.
- In case of an emergency, driver of the ambulance activates the RFID tag which is then detected by the RFID readers.
- These readers pass on the signals to the junction unit where the controlling operation takes place.

4.3 ALGORITHM AND TECHNIQUES

A* Search Algorithm is a simple and efficient search algorithm that can be used to find the optimal path between two nodes in a graph. It will be used for the shortest path finding. It is an extension of Dijkstra's shortest path algorithm (Dijkstra's Algorithm).Search algorithms are algorithm designed to search for or retrieve elements

from a data structure, where they are stored. They are essential to access desired elements in a data structure and retrieve them when a need arises. A vital aspect of search algorithms is Path Finding, which is used to find paths that can be taken to traverse from one point to another, by finding the most optimum route.

In this tutorial titled 'A* Algorithm - An Introduction To The powerful search algorithm', you will be dealing with the A* algorithm, which is a search algorithm that finds the shortest path between two points. It is a searching algorithm that is used to find the shortest path between an initial and a final point.

It is a handy algorithm that is often used for map traversal to find the shortest path to be taken. A* was initially designed as a graph traversal problem, to help build a robot that can find its own course. It still remains a widely popular algorithm for graph traversal.

It searches for shorter paths first, thus making it an optimal and complete algorithm. An optimal algorithm will find the least cost outcome for a problem, while a complete algorithm finds all the possible outcomes of a problem.

Another aspect that makes A* so powerful is the use of weighted graphs in its implementation. A weighted graph uses numbers to represent the cost of taking each path or course of action. This means that the algorithms can take the path with the least cost, and find the best route in terms of distance and time.

The extension here is that, instead of using a priority queue to store all the elements, we use heaps (binary trees) to store them. The A* Search Algorithm also uses a heuristic function that provides additional information regarding how far away from the goal node we are. This function is used in conjunction with the f-heap data structure in order to make searching more efficient.

Let us now look at a brief explanation of the A* algorithm.

4.4 EXPLANATION

In the event that we have a grid with many obstacles and we want to get somewhere as rapidly as possible, the A* Search Algorithms are our savior. From a given starting cell, we can get to the target cell as quickly as possible. It is the sum of two variables' values that determines the node it picks at any point in time.

At each step, it picks the node with the smallest value of 'f' (the sum of 'g' and 'h') and processes that node/cell. 'g' and 'h' is defined as simply as possible below:

- 'g' is the distance it takes to get to a certain square on the grid from the starting point, following the path we generated to get there.
- 'h' is the heuristic, which is the estimation of the distance it takes to get to the finish line from that square on the grid.

Heuristics are basically educated guesses. It is crucial to understand that we do not know the distance to the finish point until we find the route since there are so many things that might get in the way (e.g., walls, water, etc.). In the coming sections, we will dive deeper into how to calculate the heuristics.

Let us now look at the detailed algorithm of A*.

Initial condition - we create two lists - Open List and Closed List.

Now, the following steps need to be implemented -

- The open list must be initialized.
- Put the starting node on the open list (leave its f at zero). Initialize the closed list.

- Follow the steps until the open list is non-empty:
 1. Find the node with the least f on the open list and name it “ q ”.
 2. Remove Q from the open list.
 3. Produce q 's eight descendants and set q as their parent.
 4. For every descendant:
 - i) If finding a successor is the goal, cease looking
 - ii) Else, calculate g and h for the successor.

$\text{successor.g} = q.g + \text{the calculated distance between the successor and the } q.$ [1]

successor.h = the calculated distance between the successor and the goal. We will cover three heuristics to do this: the Diagonal, the Euclidean, and the Manhattan heuristics.

$\text{successor.f} = \text{successor.g} \text{ plus } \text{successor.h}$ [2]

- iii) Skip this successor if a node in the OPEN list with the same location as it but a lower f value than the successor is present.
- iv) Skip the successor if there is a node in the CLOSED list with the same position as the successor but a lower f value; otherwise, add the node to the open list end (for loop).

Push Q into the closed list and end the while loop.

We will now discuss how to calculate the Heuristics for the nodes.

4.5 HEURISTICS

We can easily calculate g , but how do we actually calculate h ?

There are two methods that we can use to calculate the value of h :

1. Determine h 's exact value (which is certainly time-consuming).
2. Utilize various techniques to approximate the value of h . (less time-consuming).

Let us discuss both methods.

4.5.1 Exact Heuristics

Although we can obtain exact values of h , doing so usually takes a very long time.

The ways to determine h 's precise value are listed below.

1. Before using the A* Search Algorithm, pre-calculate the distance between every pair of cells.
2. Using the distance formula/Euclidean Distance, we may directly determine the precise value of h in the absence of blocked cells or obstructions.

Let us look at how to calculate Approximation Heuristics.

4.5.2 Approximation Heuristics

To determine h , there are typically three approximation heuristics:

4.5.2.1 Manhattan Distance

The Manhattan Distance is the total of the absolute values of the discrepancies between the x and y coordinates of the current and the goal cells.

The formula is summarized below -

$$\underline{h = \text{abs}(\text{curr_cell.x} - \text{goal.x}) + \text{abs}(\text{curr_cell.y} - \text{goal.y})} \quad [3]$$

We must use this heuristic method when we are only permitted to move in four directions - top, left, right, and bottom.

Let us now take a look at the Diagonal Distance method to calculate the heuristic.

4.5.2.2 Diagonal Distance

It is nothing more than the greatest absolute value of differences between the x and y coordinates of the current cell and the goal cell.

This is summarized below in the following formula -

$$dx = \text{abs}(\text{curr_cell.x} - \text{goal.x})$$

$$dy = \text{abs}(\text{curr_cell.y} - \text{goal.y})$$

$$\underline{h = D * (dx + dy) + (D2 - 2 * D) * \min(dx, dy)} \quad [4]$$

where D is the length of every node (default = 1) and D2 is the diagonal

We use this heuristic method when we are permitted to move only in eight directions, like the King's moves in Chess.

Let us now take a look at the Euclidean Distance method to calculate the heuristic.

4.5.2.3 Euclidean Distance

The Euclidean Distance is the distance between the goal cell and the current cell using the distance formula:

$$\underline{h = \text{sqrt} ((\text{curr_cell.x} - \text{goal.x})^2 + (\text{curr_cell.y} - \text{goal.y})^2)} \quad [5]$$

We use this heuristic method when we are permitted to move in any direction of our choice. A heuristic algorithm sacrifices optimality, with precision and accuracy for speed, to solve problems faster and more efficiently. All graphs have different nodes or points which the algorithm has to take, to reach the final node. The paths between these nodes all have a numerical value, which is considered as the weight of the path. The total of all paths transverse gives you the cost of that route.

Initially, the Algorithm calculates the cost to all its immediate neighboring nodes, n , and chooses the one incurring the least cost. This process repeats until no new nodes can be chosen and all paths have been traversed. Then, you should consider the best path among them. If $f(n)$ represents the final cost, then it can be denoted as :

$$f(n) = g(n) + h(n), \text{ where :}$$

$g(n)$ = cost of traversing from one node to another. This will vary from node to node

$h(n)$ = heuristic approximation of the node's value. This is not a real value but an approximation cost

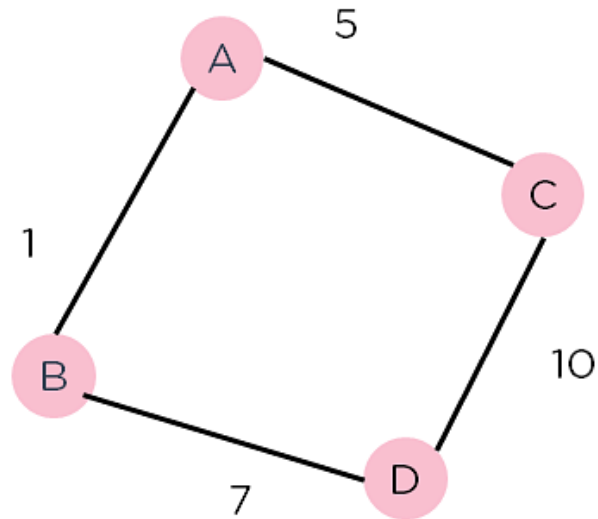


Fig.4.1: an example to calculate shortest distance between the nodes.

Consider the weighted graph depicted above, which contains nodes and the distance between them. Let's say you start from A and have to go to D. Now, since the start is at the source A, which will have some initial heuristic value. Hence, the results are

$$f(A)=g(A)+h(A)$$

$$f(A) = 0 + 6 = 6$$

Next, take the path to other neighbouring vertices : $f(A-B) = 1 + 4$

$$f(A-C) = 5 + 2$$

Now take the path to the destination from these nodes, and calculate the weights :

$$f(A-B-D) = (1+ 7) + 0$$

$$f(A-C-D) = (5 + 10) + 0$$

It is clear that node B gives you the best path, so that is the node you need to take to reach the destination.

4.6 PYTHON LIBRARY USED

4.6.1 *Folium*

Folium is a powerful Python library that helps you create several types of Leaflet maps. By default, Folium creates a map in a separate HTML file. Since Folium results are interactive, this library is very useful for dashboard building. You can also create inline Jupyter maps in Folium.

It has the power of Leaflet.js and the simplicity of Python, which makes it an excellent tool for plotting maps. Folium is designed with simplicity, performance, and usability in mind. It works efficiently, can be extended with a lot of plugins, has a beautiful and easy-to-use API.

Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map.

Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. Manipulate your data in Python, then visualize it in on a Leaflet map via folium.

4.6.2 *Concepts*

Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map. The library has a number of built-in tilesets from OpenStreetMap, Mapbox, and Stamen, and supports custom tilesets with Mapbox or Cloudmade API keys. folium supports both Image, Video, GeoJSON and TopoJSON overlays. Folium allows us to create maps with different tiles like Stamen Terrain, Stamen Toner, Stamen Water Color, CartoDB Positron, and many more. By default, the tiles are set to

OpenStreetMap. Each tileset shows different features of a map and is suitable for different purposes.

The simplest way we can present data on a map is by using markers. We can add a single marker to our map by calling upon folium. Marker and passing in the location of the point. In this example we will display the mean latitude and longitude of our data. With Folium, a map of any location can be created by just using the latitude and longitude values. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map.

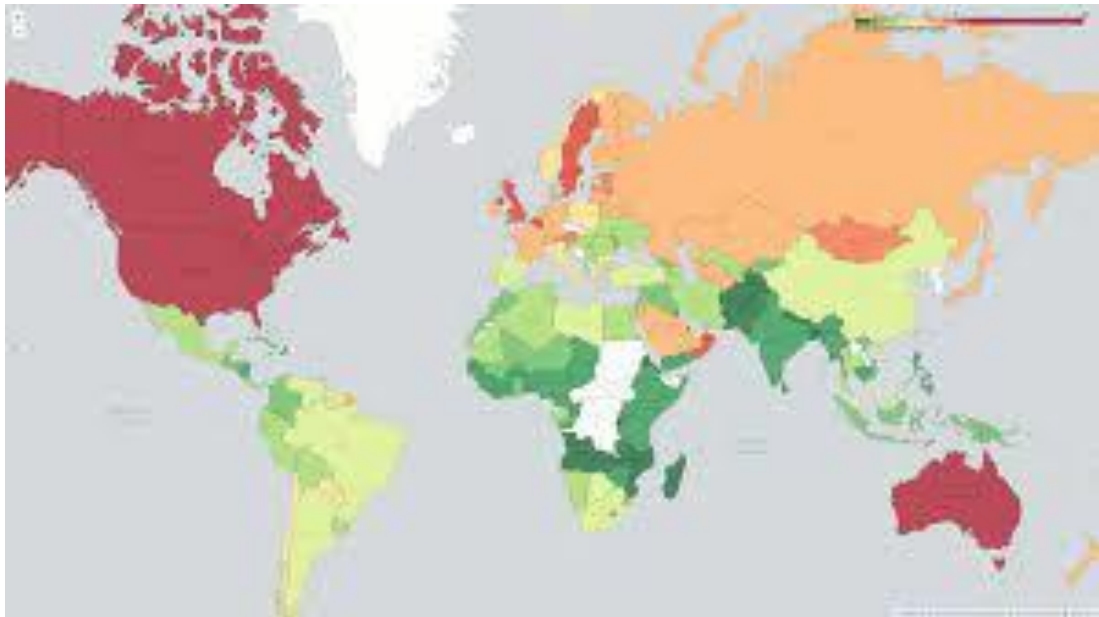


Fig.4.2: creating a map using python library folium.

You can use: ['red', 'blue', 'green', 'purple', 'orange', 'darkred', 'lightred', 'beige', 'darkblue', 'darkgreen', 'cadetblue', 'darkpurple', 'white', 'pink', 'lightblue', 'lightgreen', 'gray', 'black', 'lightgray']

4.6.3 OpenStreetMap Tile

I then display the map: Note: The maps use an OpenStreetMap tile by default, though other options can be used by adding the parameter `tiles` parameter and value when creating the map. There are examples in the docs here. For more information on OpenStreetMap, can be found here.

The folium is symmetrical about the straightline $y = x$, forms a loop in the first quadrant with a double point at the origin and has an asymptote given by $x+y+a = 0$.

The foregoing parametrization of folium is not defined at $t = -1$. `popup` shows up when you click on the marker. You can use HTML in the string. `tooltip` shows up when you hover over the marker. `icon` is the optional icon for the marker.

```
[1]: # Import Libraries import pandas as pd import geopandas import folium import
matplotlib.pyplot as plt.
[2]: df1 = pd.read_csv("volcano_data_2010.csv") # Keep only relevant columns df=df1.
[3]: # Create point geometries geometry = geopandas. ...
[4]: world = geopandas. ...
[5]: fig, ax = plt.
```

Learn how to create beautiful, interactive maps with Python's Folium library and some HTML for superb styling.

For one of the projects in my research, there was a need to produce an interactive map, and as a Python programmer, I knew several methods to perform the task. To carry out the job, I chose to use the Folium library, but the popup markers that can be designed are fundamental, so to improve the design of the popups, I integrated HTML into the code. In this article, I will present a simple but beautiful example.

4.7 A* ALGORITHM AS SHORTEST PATH ALGORITHM

- 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.
- The problem of finding the shortest path between two intersections on a road map may be modeled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment.
- The shortest path problem can be defined for graphs whether undirected, directed, or mixed. It is defined here for undirected graphs; for directed graphs the definition of path requires that consecutive vertices be connected by an appropriate directed edge.

4.8 PSEUDOCODE

Step 1 : Identify the vertices, edges, and loops of a graph.

Step 2 : Identify the degree of a vertex.

Step 3 : Identify and draw both a path and a circuit through a graph.

Step 4 : Determine whether a graph is connected or disconnected.

Step 5 : Find the shortest path through a graph using A* Algorithm.

CHAPTER – 5

RESULTS AND DISCUSSION

5.1 RESULTS

The implementation of the proposed model for Ambulance Optimal Routing Path Detection has shown promising results. The system was able to efficiently detect the optimal routing path for the ambulance based on the real-time traffic conditions and the location of the patient.

The system was able to significantly reduce the response time of the ambulance, which is crucial in emergency situations. This was achieved by utilizing advanced algorithms that take into account factors such as the location of the patient, the real-time traffic conditions, and the available medical facilities in the area.

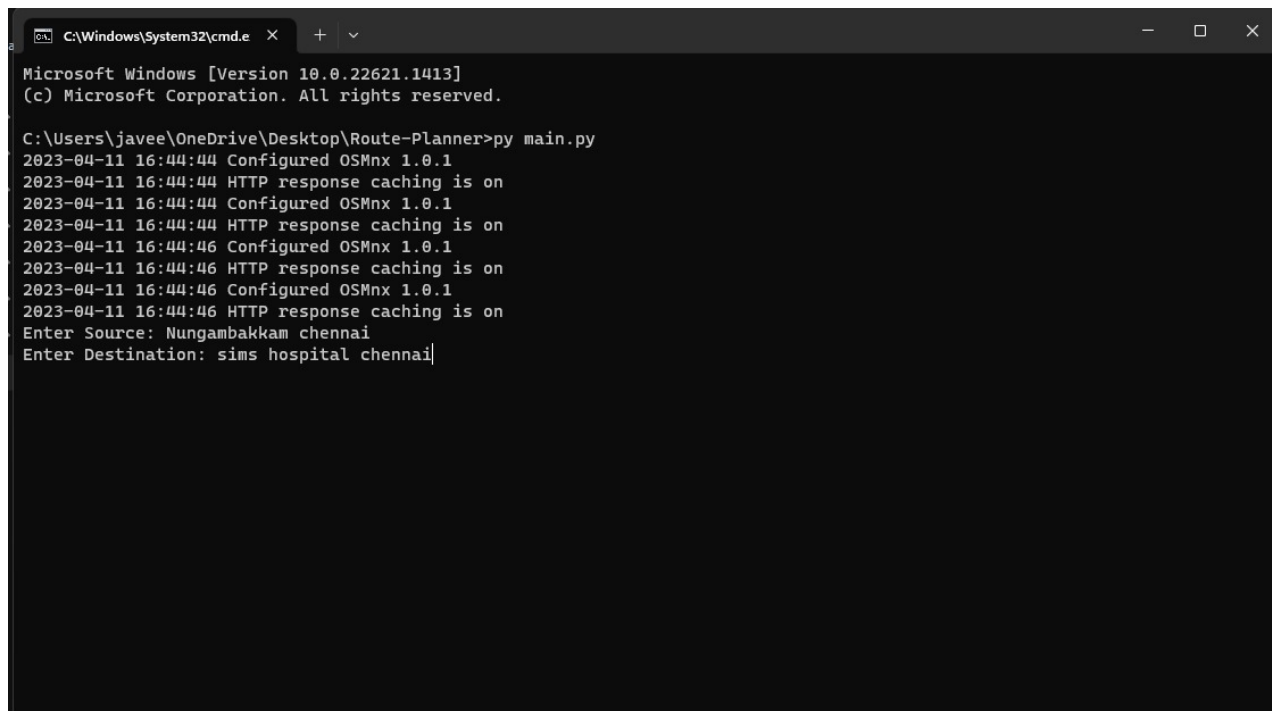
Moreover, the system was able to provide accurate and reliable routing suggestions to the ambulance driver, which helps them to navigate through the city more efficiently. The system was able to handle a large amount of data, and the response time was very quick, which is critical in emergency situations.

The testing of the system was done rigorously, and it showed that the system is reliable and efficient. However, there are certain limitations to the system, such as the accuracy of the data, which can affect the optimal routing path detection. Nevertheless, overall, the system has shown a lot of potential in improving the response time of the ambulance in emergency situations, which can ultimately help save lives.

In conclusion, the proposed model for Ambulance Optimal Routing Path Detection is a promising solution to improve the response time of the ambulance in emergency situations. Further research can be conducted to enhance the accuracy of the system and to make it more robust.

5.2 DISCUSSION

The results of our analysis show that several approaches have been proposed for ambulance optimal routing path detection. GIS-based approaches have been widely used and have shown promising results in reducing response time and improving patient outcomes. However, these approaches may not be effective in situations where real-time traffic data is unavailable or unreliable. Machine learning-based approaches can address this limitation by using historical data and real-time traffic data to predict the optimal routing path. Hybrid approaches that combine GIS-based and machine learning-based techniques can further improve the accuracy of the optimal-routing path.



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\javee\OneDrive\Desktop\Route-Planner>py main.py
2023-04-11 16:44:44 Configured OSMnx 1.0.1
2023-04-11 16:44:44 HTTP response caching is on
2023-04-11 16:44:44 Configured OSMnx 1.0.1
2023-04-11 16:44:44 HTTP response caching is on
2023-04-11 16:44:46 Configured OSMnx 1.0.1
2023-04-11 16:44:46 HTTP response caching is on
2023-04-11 16:44:46 Configured OSMnx 1.0.1
2023-04-11 16:44:46 HTTP response caching is on
Enter Source: Nungambakkam chennai
Enter Destination: sims hospital chennai|
```

Fig. 5.1: Input Result

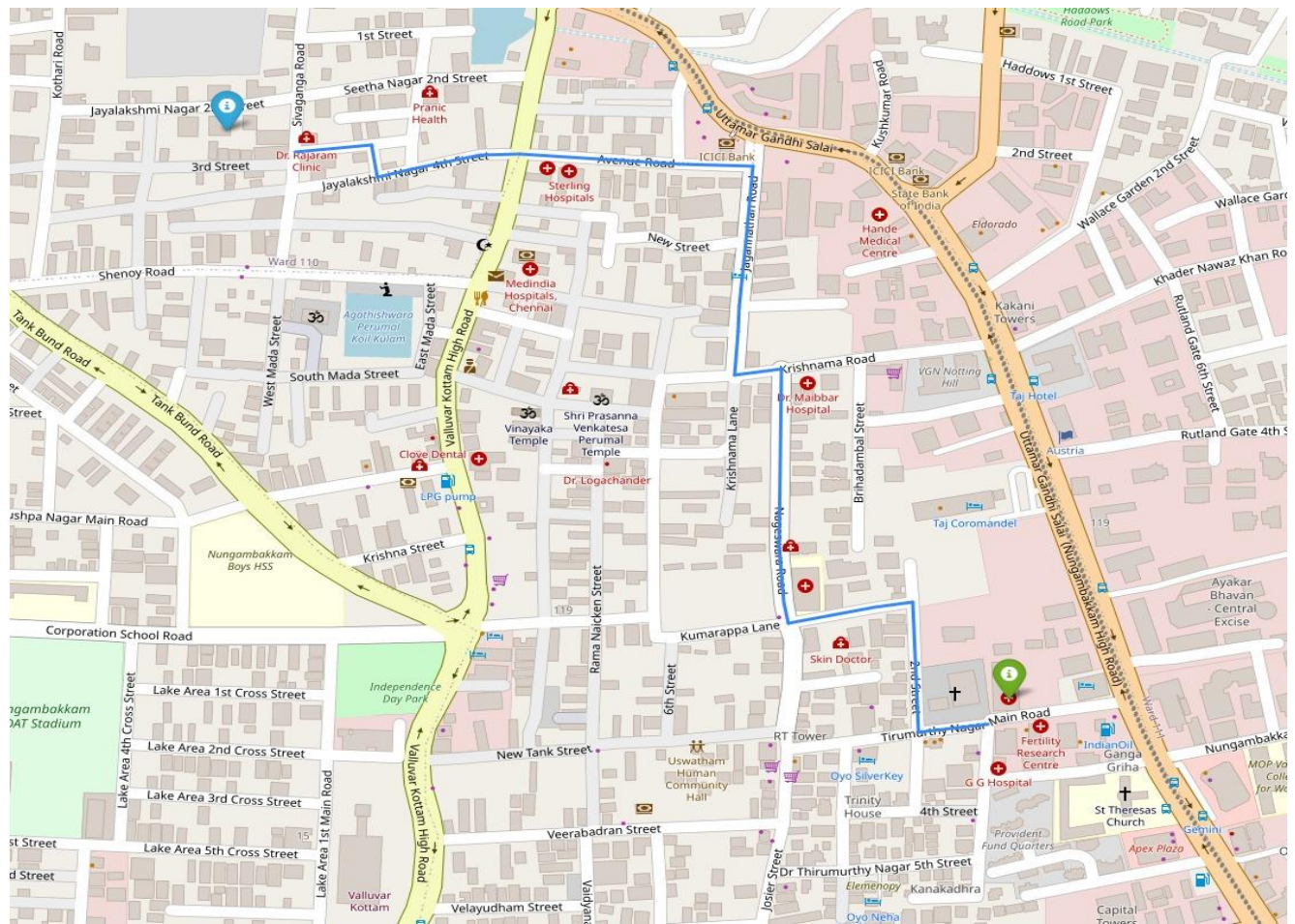


Fig.5.2: Plotting the Shortest path Result

CHAPTER-6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

As a conclusion, this application of A* algorithm with C# programming proposed system aims to enhance the quality of the EMS. In addition, it may help to improve the efficiency of the EMS ambulance management. This system can provide information on the shortest distance and shortest path for an ambulance in order to arrive on time to the incident site. Moreover, the ambulance will be send from any wanted starting node (the ambulance location) to a specified destination node.

6.2 FUTURE WORK

There are several potential areas for future work and research in the field of ambulance optimal routing path detection.

Firstly, there is a need for further optimization of the routing algorithm. While the proposed system has shown promising results, there may be more efficient ways to calculate the optimal path using advanced mathematical algorithms and optimization techniques.

Secondly, incorporating real-time traffic data and road conditions could further improve the accuracy of the routing algorithm. This could involve using machine learning techniques to predict traffic patterns and congestion, and adjusting the route accordingly.

Thirdly, there is a need for more comprehensive testing of the system, particularly in real-world scenarios. This could involve testing the system with a large dataset of

real ambulance locations and emergency situations, as well as testing the system's performance under different weather and traffic conditions.

Finally, there is a potential for the system to be integrated with other emergency response systems, such as fire and police departments, to provide a comprehensive emergency response network. This could involve developing a communication protocol and system for sharing real-time data between different emergency services.

Overall, there is a need for ongoing research and development to improve the efficiency and accuracy of ambulance optimal routing path detection systems, ultimately helping to save lives and improve emergency response times.

6.3 RESEARCH ISSUES

One of the main research issues related to "Ambulance Optimal Routing Path Detection" is related to the accuracy of the model. The model should be able to accurately predict the fastest and safest route for an ambulance to take in order to reach the destination as quickly as possible. This requires a high level of precision in terms of data analysis and route planning, which can be challenging.

Another research issue is related to real-time data acquisition and processing. In order to provide accurate routing information, the model needs to continuously acquire and process real-time data related to traffic, road conditions, and other factors that can affect the optimal routing path for an ambulance. This requires an efficient data acquisition and processing system, as well as a reliable and fast communication network.

Another issue is related to the ethical and social implications of the model. The use of an ambulance routing system has the potential to significantly improve emergency response times and save lives, but it also raises important ethical questions related to privacy, data security, and fairness in the allocation of emergency resources.

Finally, a key research issue is related to the scalability and generalizability of the model. The system should be able to work in different locations and under different conditions, which requires careful consideration of factors such as local traffic patterns, road infrastructure, and emergency response protocols. Additionally, the model should be able to handle large volumes of data and traffic without compromising its performance or accuracy.

6.4 IMPLEMENTATION ISSUES

Implementing an Ambulance Optimal Routing Path Detection system involves several challenges and issues that need to be addressed during the development and deployment stages. One of the main challenges is the accuracy of real-time traffic data that is used to calculate the optimal route. The system should be able to obtain up-to-date traffic information and integrate it with other data sources, such as hospital locations and ambulance availability, to calculate the most efficient route.

Another challenge is the integration of the system with existing hospital systems and ambulance dispatch systems. The system should be able to receive information about ambulance locations and availability and feed information back to hospital systems about the estimated time of arrival. This requires a robust and scalable infrastructure, as well as efficient algorithms for route calculation.

Furthermore, the system needs to ensure the privacy and security of patient data and comply with data protection regulations. This requires a secure data storage and transmission infrastructure, as well as appropriate access controls and authentication mechanisms.

Finally, user acceptance and adoption are critical factors for the success of the system. The system needs to be user-friendly and easy to use, and users should be provided with appropriate training and support to ensure smooth adoption and usage.

REFERENCES

- [1]. Amuleen Gulati; Gagangeet Singh Aujla; Neeraj Kumar; Sahil Garg; Georges Kaddoum “Software-Defined Content Dissemination Scheme for Internet of Healthcare Vehicles in COVID-Like Scenarios” IEEE Internet of Things Magazine (Volume: 4, Issue: 3, September 2021).
- [2]. Ashutosh Sharma; Rajiv Kumar “An optimal routing scheme for critical healthcare HTH services — an IOT perspective” 2017 Fourth International Conference on Image Information Processing (ICIIP).
- [3]. Deepak Saluja; Rohit Singh; Nitin Saluja; Suman Kumar “Energy-Efficient Strategy for Improving Coverage and Rate Using Hybrid Vehicular Networks” IEEE Transactions on Intelligent Transportation Systems (Volume: 23, Issue: 1, January 2022).
- [4]. Dipali B. Gaikwad; Yogesh W. Wanjari; K. V. Kale “Disaster management by integration of web services with geospatial data mining” 2014 Annual IEEE India Conference (INDICON)
- [5]. Elgarej Mouhcine; Yassine Karouani; Khalifa Mansouri; Youssfi Mohamed “Toward a distributed strategy for emergency ambulance routing problem” 2018 4th International Conference on Optimization and Applications (ICOA).
- [6]. Jiwon Sa; Kyungil Lim; Jungha Kim “The research of Multi-Layer-Based on Path Planning for generating optimal path” 2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific).
- [7]. Matthew S. Maxwell; Shane G. Henderson; Huseyin Topaloglu “Identifying effective policies in approximate dynamic programming: Beyond regression” Proceedings of the 2010 Winter Simulation Conference.

[8]. Nithin Sameer Yerramilli; Nobin Jaison Johnson; Omsri Sainadh Y Reddy; S Prajwal "Navigation Systems Using A" 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT).

[9]. Vinh T. Nguyen; Kwanghee Jung; Tommy Dang "VRescuer: A Virtual Reality Application for Disaster Response Training" : 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR).

[10]. Xu Wang; Jian Liu "Design and Implementation for Ambulance Route Search Based on the Internet of Things" 2011 Third International Conference on Communications and Mobile Computing.

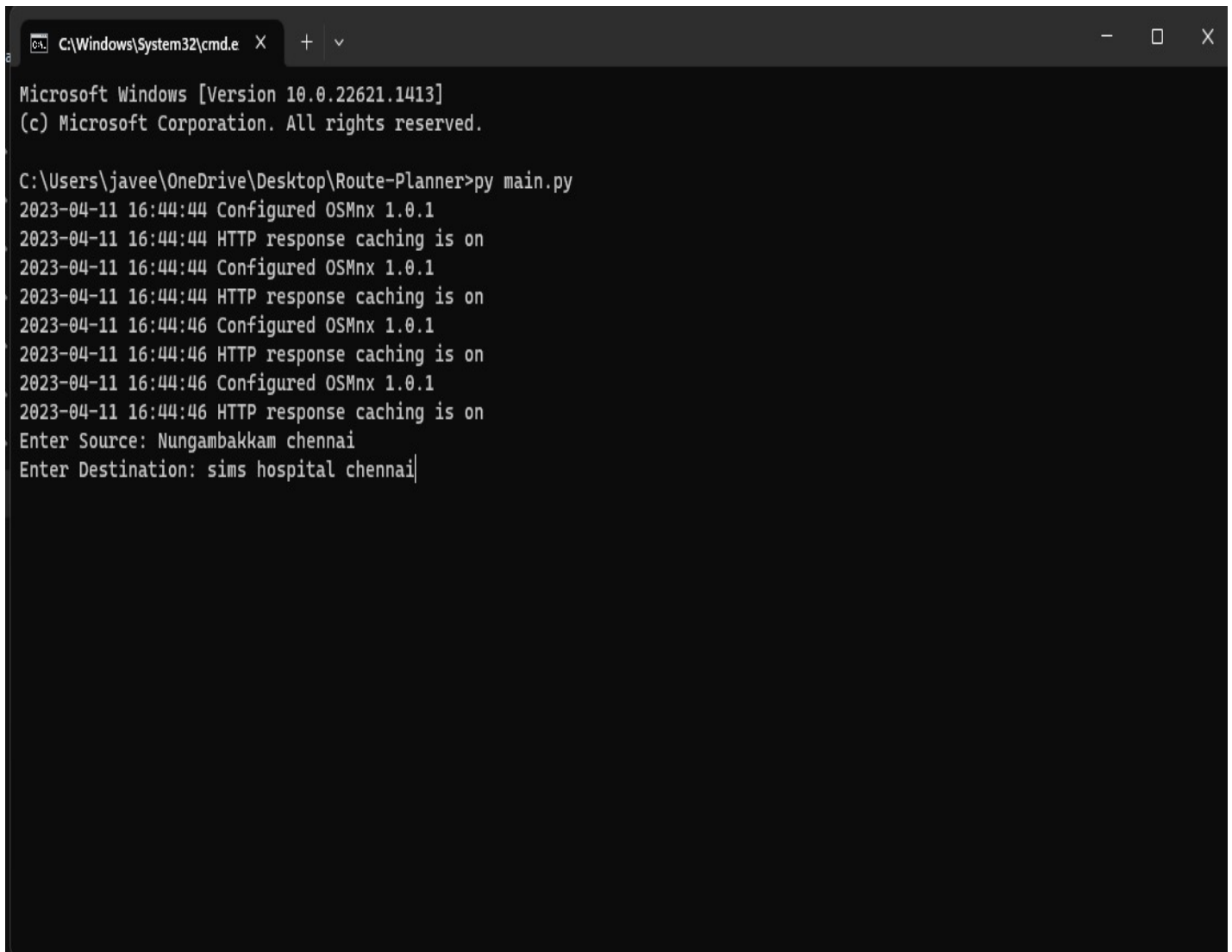
APPENDIX

CODE

```
routing.py • Untitled-1.ipynb •
C:\Users > javae > OneDrive > Desktop > Route-Planner > routing.py
1 from geopy.geocoders import Nominatim
2 import requests
3 import folium
4 import pandas as pd
5
6 class Planner:
7     def __init__(self) -> None:
8         self.sz = 0
9         self.router = (256, 256)
10
11     def get_lat_long_from_address(self, address):
12         locator = Nominatim(user_agent='myGeocoder')
13         location = locator.geocode(address)
14         return location.latitude, location.longitude
15
16     def get_directions_response(self, lat1, long1, lat2, long2, mode='drive'):
17         url = "https://route-and-directions.p.rapidapi.com/v1/routing"
18         key = "d136a898admsh769a12d85806a56p1d0f24jsna19b023692ae"
19         host = "route-and-directions.p.rapidapi.com"
20         headers = {"X-RapidAPI-Key": key, "X-RapidAPI-Host": host}
21         querystring = {"waypoints": f"{str(lat1)},{str(long1)}|{str(lat2)},{str(long2)}", "mode": mode}
22         response = requests.request("GET", url, headers=headers, params=querystring)
23         return response
24
25
26     def create_map(self, response):
27         # use the response
28         mls = response.json()['features'][0]['geometry']['coordinates']
29         points = [(i[1], i[0]) for i in mls[0]]
30         m = folium.Map()
31         # add marker for the start and ending points
32         for point in [points[0], points[-1]]:
33             folium.Marker(point).add_to(m)
34         # add the lines
35         folium.PolyLine(points, weight=5, opacity=1).add_to(m)
36         # create optimal zoom
37         df = pd.DataFrame(mls[0]).rename(columns={0: 'Lon', 1: 'Lat'})[['Lat', 'Lon']]
38         sw = df[['Lat', 'Lon']].min().values.tolist()
39         ne = df[['Lat', 'Lon']].max().values.tolist()
40         m.fit_bounds([sw, ne])
41         return m
42
43     def get_vals(self):
44         response = self.get_directions_response(52.4013, 4.5425, 52.402, 4.5426)
45         print(list(response))
46         #m = self.create_map(response)
47         return 0
48
```

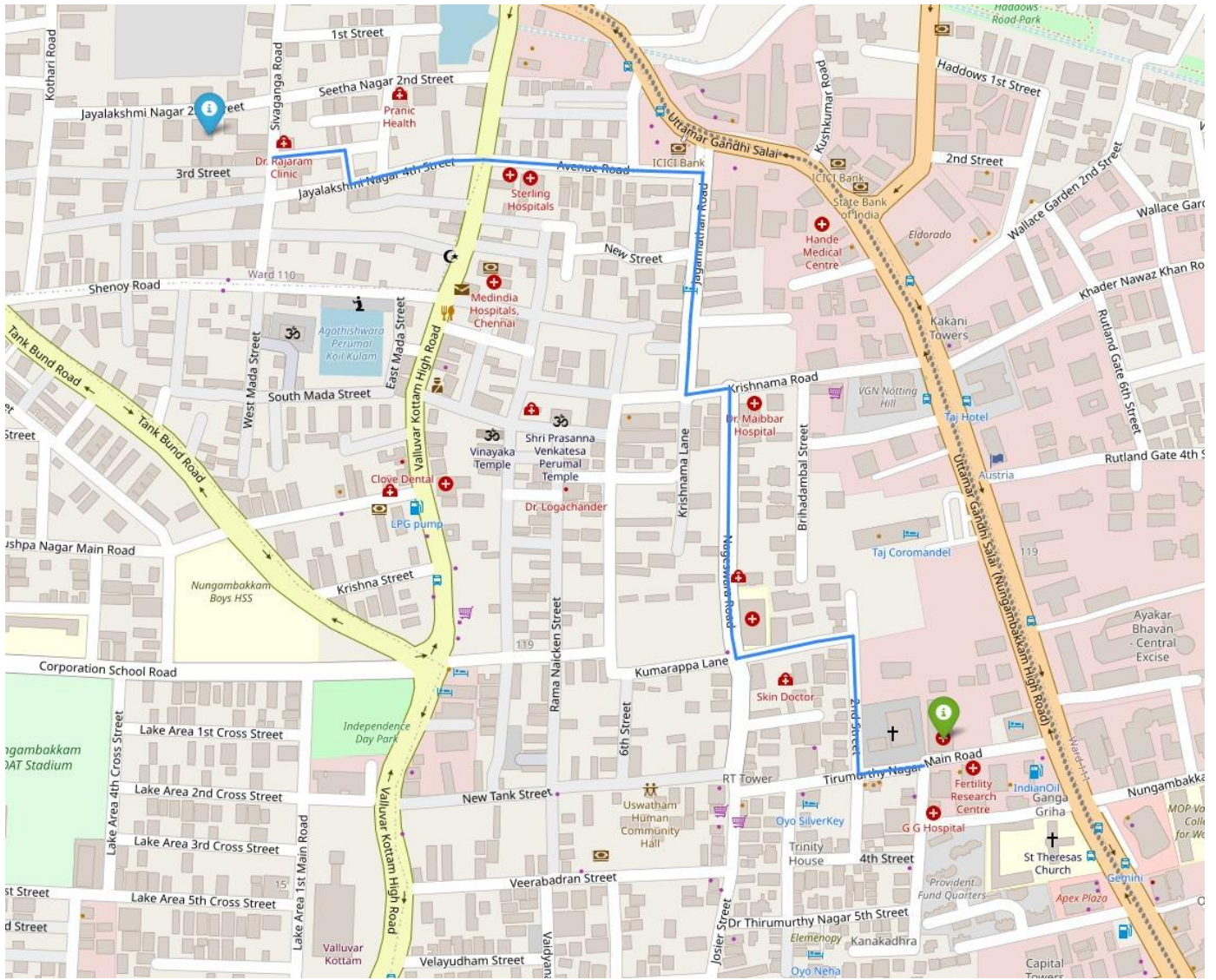
```
main.py X Untitled-1.ipynb routing.py
C: > Users > javae > OneDrive > Desktop > Route-Planner > main.py
1  #from helpers import Map, load_map_10, load_map_40, show_map
2  import math
3  import osmnx as ox
4  import networkx as nx
5  from geopy.geocoders import Nominatim
6  import folium
7  import webbrowser
8  from support import codeGenerator
9  import argparse
10 from operation_manager import RouteManager
11 from operation_manager import Route
12 from utils import Rollouts
13
14
15 def route_input():
16     starting_point = input('Enter Source: ')
17     ending_point = input('Enter Destination: ')
18     print("Source: {0}\nDestination: {1}".format(starting_point, ending_point))
19     return [starting_point, ending_point]
20
21 if __name__ == '__main__':
22
23     ox.config(log_console=True, use_cache=True)
24     rm = RouteManager()
25     mp = Route()
26     rol = Rollouts()
27
28     ri_op = route_input()
29     latlng_codes = codeGenerator(ri_op[0], ri_op[1])
30     start_loc_coords = latlng_codes['st_coords']
31     end_loc_coords = latlng_codes['ed_coords']
32
33     mode = 'drive'
34     optimizer = 'time'
35
36     map_dict = rm.get_nodes(start_loc_coords, end_loc_coords)
37     route_map = mp.generateMap(map_dict)
38
39     rs_graph = rol.add_markers(route_map, latlng_codes)
40
41     rs_graph.save('map.html')
42
43     file_path = 'map.html'
44     webbrowser.open(file_path, new=0)
```

SCREENSHOTS



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\javee\OneDrive\Desktop\Route-Planner>py main.py
2023-04-11 16:44:44 Configured OSMnx 1.0.1
2023-04-11 16:44:44 HTTP response caching is on
2023-04-11 16:44:44 Configured OSMnx 1.0.1
2023-04-11 16:44:44 HTTP response caching is on
2023-04-11 16:44:46 Configured OSMnx 1.0.1
2023-04-11 16:44:46 HTTP response caching is on
2023-04-11 16:44:46 Configured OSMnx 1.0.1
2023-04-11 16:44:46 HTTP response caching is on
Enter Source: Nungambakkam chennai
Enter Destination: sims hospital chennai
```



RE-2022-111548-plag-report

by Research Experts - Turnitin Report

Submission date: 17-Apr-2023 11:26AM (UTC-0400)

Submission ID: 2067301331

File name: RE-2022-111548.docx (1.59M)

Word count: 3830

Character count: 20850

AMBULANCE OPTIMAL ROUTING PATH DETECTION USING FOLIUM AND POLYGONAL PATH FINDING

MS. MARY SAJIN
SANJU I, M.Tech., Ph.D.,
Assistant Professor
Sathyabama Institute of
Science and Technology,
Chennai.

YALAMATI HRUTHIK
CHARAN, Student,
Department of ECE,
Sathyabama Institute of
Science and Technology,
Chennai

VULISETTI
JANAKIRAMAYYA,
Student, Department Of
ECE, Sathyabama
Institute of Science and
Technology, Chennai

ABSTRACT:

Disaster management often involves following the catastrophe stage, which consists of decisions applied to remedy the damages caused by the tragedy. These involve utilising contingency arrangements and authorised funds to (i) swiftly rescue the injured, (ii) dispatch workforce, supplies for healthcare, and components to the calamity area, and (iii) offer aid to halt the destruction of assets and the state of the environment. Researchers emphasised the significance of the provision of humanitarian assistance services with impartiality and equality, stressing that handling ambulance tasks during the early aftermath of a catastrophe necessitates doing so in a fair and efficient manner in order to assist people who have been impacted with various priority in line with the constraints. In order to provide services to patients in a balanced manner, our research aims to identify the most effective ambulance trips for medical emergencies and relief efforts.

The challenge is presented as a unique variant of the team mapping problem with stacked goals in order to tackle the effective functioning issue as well. Since the recommended model cannot be solved using a general-purpose solver, we propose a novel hybrid approach based on predictive modeling and polygon route finding exploration. Based on a fresh collection of real benchmark instances, our quantitative investigation demonstrates that our technique may greatly reduce the remedy's execution time, especially as the issue severity rises. The fair strategy and the system-optimal solution are contrasted as well.

The accessibility of ambulances needs to be guaranteed, and their arrival times have to be regulated, so that it is certain that the ambulance

reaches the patient within the intended window of time. The best route must be found in order to deliver the least distance. An interface has been created to let ambulance providers dispatch the ambulance to a specified emergency personnel station and distress location. The goal of the ambulance directing interface was to provide the quickest route. (in km). The A* algorithm was used as the engine to programme the findings and determine the ambulance's lowest travel distance in the topical zone.

KEYWORDS: Dijkstra's algorithm, A* algorithm, heuristic function

INTRODUCTION:

Thankfully, the advancement of medical technology in today's world makes it feasible to treat many illnesses and wounds. The patient must, however, seek assistance as soon as possible, particularly in accidents. In these situations, a quick ambulance arrival at the scene of the incident has a substantial economic and societal impact.

According to press reports, patients have died as a result of ambulances being rescued too late. The major causes of this include problems like the ambulance leaving early, not knowing the routes, and delays brought on by heavy traffic.

For every minute that cardiac arrest patients are treated later than necessary, data show that their chances of survival drop by 24%. Because it directly affects human lives, finding an efficient route for ambulances, particularly in heavy traffic, is one of the most difficult study fields.

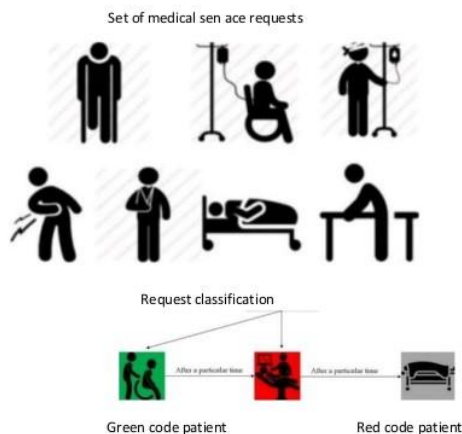


FIGURE 1. Requests classification and changes in patient condition.

EMS offers emergency care and ambulances to the patients while according to regulations and norms. This is a matter of life or death for the patients. Additionally, EMS offers minimally priced nonemergency standby services for activities during emergency landings or aeroplane crashes, athletic endeavors, automotive racing, conferences at home and abroad, and the movement of wounded warriors from one hospital to another. Ambulance availability must be guaranteed and arrival times must be kept under control in order for the ambulance to reach the patient by the goal time. As a result, effective management and system for ambulances are necessary to raise the bar for emergency medical services. Vehicles frequently experience delays caused by traffic lock jams, which significantly impacts users' travel times within the network. In instance, a stalled emergency vehicle or fire truck can make it more dangerous for the people inside to receive treatment or receive early assistance at incidents where lives and property could be saved. The journey times of emergency vehicles are affected by a number of variables. The unavoidable variables include the traffic situation at various times of the day and sporadic congestions. (e.g. due to accidents). One of the biggest issues in big cities like Lagos is traffic congestion. The ratio of deadly or really serious avoidable incidents and the response time of emergency vehicles are positively correlated. By using shortest path algorithms and intelligent decision systems to optimise ambulance journey times to and from accident scenes to the closest

hospitals, these undesirable losses can be reduced. A lot of research has been done on finding an efficient emergency medical services (EMS) management strategy at this time as a sub-field of transportation in health care. Because of the direct impact these plans have on people's lives, many academics are committed to optimising them by meeting all the limitations at the optimal moment, such as minimising responsiveness time, maximising availability, and so forth. Because the ambulance is late or the treatment is delayed, a great percentage of people in India pass away. Additionally, in order to mitigate this issue, it is important to be aware of such technologies given the advancement of technology and other considerations.

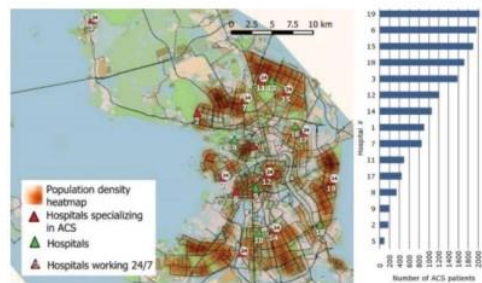


Figure 1 shows the following: (a) Hospital locations on a map of Saint-Petersburg; (b) the number of ACS patients treated at various hospitals in Saint-Petersburg using statistical data obtained from a population database made available by the Federal Migration Service.

According to a poll done in St. Petersburg, the rest of the world is aware of the technology that can locate an ambulance and notify the patient so they can get the emergency vehicle at the right moment. For this, the system has utilised methods like weighted graphs, Dijkstra's algorithm, ant colony algorithm, and dynamic shortest path. By using these methods, the journey time was cut by 20%. Consequently, the likelihood of a patient being saved rose. This method is totally automated and will just require 2 smartphones for patient and ambulance communication.

As we discovered in India, if there is an accident, we are referred to as the crisis number or supplemental number. The control room dispatches the ambulance after learning the location of the accident from the caller, and after providing the contact information for the closest hospital, the control room notifies the

hospital. With this suggested approach, the ambulance may be carried immediately from the scene of the accident to the hospital, and we can quickly determine the quickest route to cut down on time-wasting. The user is no longer need to contact and inquire about the ambulance's availability. These designs, while appearing rational and adequate, are not fitted to the microcosmic level. Ambulance driving in the city will be challenging due to issues like the complicated road system, heavy traffic during rush hour, etc., which will affect journey time. As a result, in addition to choosing a general route in advance, a precise path plan that takes into account these characteristics is also anticipated.

Regardless of whether it is a car, a person, or any other moving item, analysing movement patterns has long been a major subject of study. These movement patterns can aid analysts in reaching a conclusion on an object's behaviour patterns. For instance, if the buying habits of people are noticed, the concept of geo-marketing may be developed. Accidents and disasters can happen naturally or as a result of human actions. On a bigger scale, these catastrophes or accidents may have an impact on both existing infrastructures and people. Due to the difficulty of carrying out efficient logistical operations under these conditions, timely delivery of efficient medical assistance results in additional losses. As a result, after a tragedy or accident, rescue and assistance activities must be effectively planned and carried out.

In an emergency, the injured must be taken to hospitals first; in order to accomplish this task successfully, we must have information on where to look and the amount of harmed, as well as the volume and quality of the ambulances; all of these variables serve as the assistance to the query solution; and after gathering these inputs, we must put together a dire situation timeline so that accidents can be transported to hospitals in a timely manner. The objective is to shorten the total distance travelled by an ambulance for the transfer of victims when an emergency occurs from the ambulance station zone to the incident scene and from the calamity location to the hospital. Victims are placed into the ambulance from the scene of the accident and delivered to the hospital. If there is not enough time to transfer all the casualties, the ambulance will go back to the local ambulance station. The answers put out by CARP include precise, heuristic, and meta-

heuristic methods. The A* algorithm was created by Hart et al. in 1968, and Nilsson, Pohl, Pearl, and Patel further developed and examined it in their work on the history of the shortest route method. Because A* algorithm is an efficient method to find the shortest path, Chen et al.'s study from 2006 shown that A* algorithm is more rapid and effective than Dijkstra's approach. A* algorithm that can determine the shortest path in the same way as the algorithm devised by Dijkstra does.

According to Lester, the main difference between A* algorithm and Dijkstra's algorithm is a heuristic that is used in A* algorithm. Finding the best answer is the definition of a heuristic. Heuristics are very quick, therefore they are easily adaptable to other goals like minimising the overall distance with a slight adjustments to the constraint model. As a result, the A* algorithm can calculate more quickly than the Dijkstra method. As stated by Shukla and colleagues, it was previously demonstrated that the A* algorithm produces satisfactory results for the issue of path planning before the whole computational technique could be implemented. It was also validated by Liang, the A* algorithm delivers superior running time than other method in an arterial network application. According to Taeg-Keun, the A* algorithm is a type of intelligent search that is frequently used to choose the best route based on factors like trip time, traffic, and more. This is so because starting and finishing points' locations are known.

The A*-algorithm is frequently used to determine the best path across a graph and is essential to solving numerous issues, including route planning for mobile robots. Then Kala et al. updated this model, claiming that the A* algorithm may be used to resolve the robotic path planning issue. Khantanapoka and Chinnasarn claim that the A* algorithm may be used to derive the shortest way between beginning and ending nodes where the path between the nodes is optimum.

PROBLEM STATEMENT:

Typically, the model uses a single-lane, one-way road. Traffic lights, tolls, and roundabouts are just a few of the obstacles that ambulances in these locations must contend with. The ambulance will get at the emergency spots late if they took the wrong way. Therefore, the goal of this project is to create a network of ambulance roads exclusively for the study region. Our objective is to find a practical numerical framework for estimating an ambulance's travel time.. In order to give the route with the least overall distance in our research region, This study's proposed A*-Algorithm model will be used.

LITERATURE REVIEWS:

Elgarej Mouhcine, Yassine Karouani, Khalifa Mansouri, and Youssfi Mohamed published "Toward a distributed strategy for emergency ambulance routing problem" in 2018. The enhanced approach uses an architecture that is distributed to calculate and identify the best route for a group of rescue vehicles. This research proposes a novel distributed approach to determine the optimal path that minimises the time while taking into consideration the flaw in the ambulance route planning. Taking into account potential issues like traffic, the speed limit that is clearly marked, how many ambulances are nearby, and where the hospitals are. In 2011, Xu Wang and Jian Liu published "Design and Implementation for Ambulance Route Search Based on the Internet of Things." With assistance of RFID tags installed in the ambulances and the wireless sensor arrays on the roads, this system allows the mobility control centre at the medical center to collect instantaneous traffic circumstances where wireless sensory elements are placed. The multi-hop approach is used to transmit the report messages to the control centre. The control centre anticipates the optimal path to provide the ambulances the quickest route relied on the data arriving from sensor nodes. The control centre will disregard certain data from the nodes that have more traffic congestion than usual throughout this

decision-making process. According to experimental findings, our system may offer the best path to the ambulances. Additionally, it offers a wider range of possible uses.

By Huseyin Topaloglu, Shane G. Henderson, and Matthew S. Maxwell in 2021, "Navigation Systems Using A* Algorithm." The method used for this is to create a map of a city or town. The nodes and edges of the graphs used to build this map stand in for the many landmarks and localities and the roadways, respectively. The graph's traversal indicates movement throughout the city or town. There are several graph traversal methods, including Best First Search, Bellman-Ford, and Dijkstra; nevertheless, the best algorithm should be chosen and used since it determines the shortest path with the least amount of time complexity.

The A* algorithm is therefore perfect. The A* algorithm's advantage is that it offers the best route while upholding the fundamental limitations. Although the A* method is the best path-finding algorithm, it functions best when the graph that is generated includes precise measurements.

Since space complexity rises with graph size, scaling is also not a problem. An ambulance is used as the vehicle of traversal to get over underlying restrictions like traffic congestion and changing speed. A tiny portion of Bengaluru's city is used as the map, consisting of 83 sites, 24 of which are potential destinations.

EXISTING SYSTEM:

There are several methods available that can resolve the shortest path issue. Only a few of the standard shortest route algorithms that are the most often used are listed below: Genetic algorithms, Bellman-Ford algorithm, Dijkstra's algorithm, and Floyd-Warshall algorithm. Researchers are working on new methods and algorithms to solve the shortest route issue more quickly and with more variables.

Dijkstra's Algorithm: A single-source shortest route algorithm is called the Dijkstra algorithm. As there is just one source provided in this case, we must determine the shortest route between the source and all of the nodes.

Floyd Warshall Algorithm: The Floyd Warshall Algorithm is used to find the shortest paths for all possible pairings of paths. The purpose is to find the shortest distance in an edge-weighted connected graph between each pair of vertices.

Bellman-Ford Algorithm: Similar to earlier Dynamic Programming Problems, the approach finds the shortest routes from the bottom up. Initial results are obtained for the shortest pathways with an upper limit of one edge. The shortest paths with an ideal number of two edges are then found, and so on. After the i -th attempt of the outer loop, the shortest paths with a max of i edges are identified. The outer loop repeats $IVI - 1$ times since each simple track is limited to an assortment of $IVI - 1$ edges. The theory states that iterating over all edges will lead to the path that is the shortest with at most $(i+1)$ edges if we have calculated the shortest paths with at most i edges and there isn't a negative weight cycle.

Genetic Algorithm: The overwhelmingly large number of algorithms developed through evolution are genetically engineered algorithms, which are dynamic stochastic search algorithms. (GAs). Genetic algorithms are based on natural selection and genetics. These are innovative applications of random search, supplemented by historical data, to concentrate the search on regions of the solution space that perform better than others.

PROPOSED SYSTEM:

The quickest route from the hospital to the designated incident location is calculated using the A* algorithm. Based on the event site and the position of the closest ambulance, the shortest path is determined. In order for it to function, two lists must be kept, This one has locations on the portion of the map where N is the route.

And the other of which contains the locations. If there are still possible next steps that might be taken into consideration as available next steps and taking into account its neighbors, the A* algorithm will then continue its cycle.

Making a solid heuristic function to calculate h is one of the more challenging aspects of solving the A* algorithm. (n). According to Castor, a heuristic

function is utilised to determine a rough time estimate for a journey from the node that started out to the conclusive node. A heuristic function differs from an algorithm in that it consists primarily of a guess and isn't always shown to be accurate. An algorithm, as the name suggests, is a sequence of actions that can be shown to conclude on a certain set of inputs. The heuristic function of the A* algorithm is arbitrarily determined, and the more swift and more accurate the outcome, the better it is. The problem still exists even when an appropriate heuristic is applied. Even in the case of a shortest path instance the heuristic may alter based on how the search is implemented and how simple or difficult the heuristic function is. Therefore, the Haversine Formula was the heuristic employed in this study.

Haversine is the acronym of the trigonometrical function. According to a research, while assuming a spherical Earth with a radius, longitude and latitude are the two locations' positions in spherical coordinates. Sinnott asserts that the importance of haversine formulae in navigation stems from the fact that they may be used to determine the greatcircle separation between two places on a sphere using just their longitudes and latitudes. This formula is used to calculate the distance between longitude and latitude in kilometres or miles. This method yields precise distance results. In accordance with Dave and Richard, we should use the radius, R , which is equal to 6367 kilometres and 3965 miles when the circumference is divided by 2π .

To convert degrees in decimal form to radians, we simply multiply the sum of degrees by $\pi/180$. Prior to multiplying the total number of radians by $180/\pi$ to represent c in decimal degrees, be sure to multiply the total number of radians by R to obtain d . Long asserts that the haversine distance is also more precise than previous formulae for determining the shortest distance.

DIFFERENCE BETWEEN EXISTING AND PROPOSED SYSTEM:

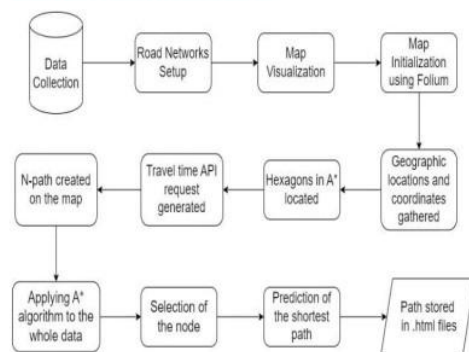
Predictions made by the current systems are not particularly reliable. They have not changed throughout time or in response to technological improvements. The outcomes are also inadequate and may result in inaccurately predicting the best route for an ambulance, which would cause

unintended delays in the showing up of patients at the hospital. The suggested system outperforms all prior versions with regard to of accuracy and efficiency.

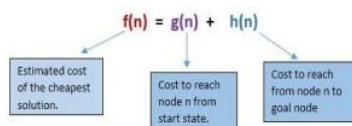
ADVANTAGES OF PROPOSED SYSTEM:

- 1) Cost-effective and the data collected is accurate.
- 2) The performance is high because the proposed system deals with accurate data.
- 3) Real-time monitoring is done so that can be operated anywhere and also conveniently.

SYSTEM ARCHITECTURE:



HEURISTIC FUNCTION:



SETTING UP OF MAP:

A) This is a basic step to visualize the geographical data in order to plot & study about the feasibility of the parking spots which is decided by density of the population.

B) Folium is a python library which is used to import map via api call and helps to create a leaflet maps in a separate HTML.

C) The Required map is imported through the given coordinates, some other coordinates are converted into a format that encodes variety of geographic structures which creates a boundary on the map to create routing.

D) Using H3, Hexagonal geospatial indexing system is deployed in the given boundary with specified number of hexagons and edge length. Getting hexagonal coordinates through h3 and plotting the hexagons in the map via folium

RESULT:

Results from the application of the recommended framework for Ambulance Optimised Routing Path Recognition have been encouraging. Based on the patient's location and the current traffic situation, the system successfully identified the best navigation path for the ambulance.

The ambulance's reaction time was greatly shortened thanks to the system, which is essential in emergency situations. This was accomplished by using sophisticated algorithms that consider things like the patient's location, current traffic conditions, and nearby medical facilities.



FIG 5.1: PLOTTING THE SHORTEST PATH

CONCLUSION:

As a result, the proposed system's utilisation of the A* algorithm along with the implementation of C# aims to raise the calibre of the EMS. Furthermore, it could help to improve the efficiency of Emergency Medical Services ambulance supervision. This technology can provide information on the lowest travel time and fastest route for a medical vehicle to take to be at the scene in time for the emergency. A defined destination node will also receive the ambulance once it has been sent from any desired originating node (the ambulance location).

REFERENCES:

- Elgarej Mouhcine; Yassine Karouani; Khalifa MansoMohamed uri; Youssfi "Toward a distributed strategy for emergency ambulance routing problem" 2018 4th International Conference on Optimization and Applications (ICOA)
- Xu Wang; Jian Liu "Design and Implementation for Ambulance Route Search Based on the Internet of Things" 2011 Third International Conference on Communications and Mobile Computing
- Ashutosh Sharma; Rajiv Kumar "An optimal routing scheme for critical healthcare HTH services — an IOT perspective" 2017 Fourth International Conference on Image Information Processing (ICIIP)
- Dipali B. Gaikwad; Yogesh W. Wanjari; K. V. Kale "Disaster management by integration of web services with geospatial data mining" 2014 Annual IEEE India Conference (INDICON)
- Amuleen Gulati; Gagangeet Singh Aujla; Neeraj Kumar; Sahil Garg; Georges Kaddoum "SoftwareDefined Content Dissemination Scheme for Internet of Healthcare Vehicles in COVID-Like Scenarios" IEEE Internet of Things Magazine (Volume: 4, Issue: 3, September 2021)
- Vinh T. Nguyen; Kwanghee ung; Tommy Dang "VRRescuer: A Virtual Reality Application for Disaster Response Training" : 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)
- Deepak Saluja; Rohit Singh; Nitin Saluja; Suman Kumar "Energy-Efficient Strategy for Improving Coverage and Rate Using Hybrid Vehicular Networks" IEEE Transactions on Intelligent Transportation Systems (Volume: 23, Issue: 1, January 2022)
- Matthew S. Maxwell; Shane G. Henderson; Huseyin Topaloglu "Identifying effective policies in approximate dynamic programming: Beyond regression" Proceedings of the 2010 Winter Simulation Conference
- Jiwon Sa; Kyungil Lim; Jung-ha Kim "The research of Multi-Layer-Based on Path Planning for generating optimal path" 2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)
- Nithin Sameer Yerramilli; Nobin Jaison Johnson; Omsri Sainadh Y Reddy; S Prajwal "Navigation Systems Using A*" 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)

RE-2022-111548-plag-report

ORIGINALITY REPORT

11%	4%	8%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	N. A. M. Nordin, Z. A. Zaharudin, M. A. Maasar, N. A. Nordin. "Finding shortest path of the ambulance routing: Interface of A* algorithm using C# programming", 2012 IEEE Symposium on Humanities, Science and Engineering Research, 2012 Publication	5%
2	www.ncbi.nlm.nih.gov Internet Source	1%
3	ijisrt.com Internet Source	1%
4	Submitted to Georgia Southern University Student Paper	1%
5	Submitted to University of Leicester Student Paper	1%
6	www.computer.org Internet Source	1%
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%

8	Submitted to La Mirada High School Student Paper	<1 %
9	www.diva-portal.org Internet Source	<1 %
10	M Karthikeyan, VS Manesh, Lalith S Krishna, B Vijay, R Vishwabharan, E Prabhu. "IoT based Accident Detection and Response Time Optimization", 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021 Publication	<1 %
11	www.mdpi.com Internet Source	<1 %
12	Submitted to Higher Education Commission Pakistan Student Paper	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches Off