PATH OPTIMIZATION: A POTENTIAL APPROACH TO PATH PLANNING FOR VEHICLE NAVIGATION

A Dissertation

Submitted

In Partial Fulfillment of the Requirements for

The Degree of

MASTER OF TECHNOLOGY

In

Computer Science & Engineering

Submitted by:

Anum Kamal
Enroll No. 1400100477
Roll No. 18001621003

Under the Supervision of:

Dr. Faiyaz Ahmad

(Assistant Professor)

Department of Computer Science & Engineering

Faculty of Engineering

INTEGRAL UNIVERSITY, LUCKNOW, INDIA

August, 2020

CERTIFICATE

This is to certify that **Ms. Anum Kamal** (Enroll. No. 1400100477) has carried out the research work presented in the dissertation titled **"Path Optimization: A Potential Approach to Path Planning for Vehicle Navigation"** submitted for partial fulfillment for the award of the **Master Of Technology In Computer Science & Engineering** from **Integral University, Lucknow** under my supervision.

It is also certified that:

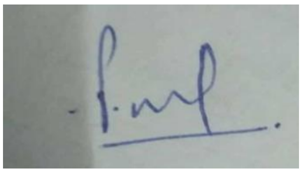
(i) This dissertation embodies the original work of the candidate and has not been earlier submitted elsewhere for the award of any degree/diploma/certificate.

(ii) The candidate has worked under my supervision for the prescribed period.

(iii) The dissertation fulfills the requirements of the norms and standards prescribed by the University Grants Commission and Integral University, Lucknow, India.

(iv) No published work (figure, data, table etc) has been reproduced in the dissertation without express permission of the copyright owner(s).

Therefore, I deem this work fit and recommend for submission for the award of the aforesaid degree.



Dr. Faiyaz Ahmad
Dissertation Guide
(Assistant Professor)
Department of CSE,
Integral University, Lucknow

Dr. Mohammadi Akheela Khanum
H.O.D.
Department of CSE,
Integral University, Lucknow

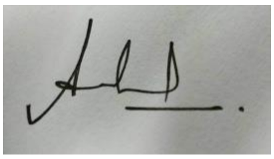
Date: 08 / 08 / 2020
Place: Lucknow

DECLARATION

I hereby declare that the dissertation titled " Path Optimization: A Potential Approach to Path Planning for Vehicle Navigation" is an authentic record of the research work carried out by me under the supervision of Dr. Faiyaz Ahmad, Department of Computer Science & Engineering , for the period from August, 2019 to August, 2020 at Integral University, Lucknow. No part of this dissertation has been presented elsewhere for any other degree or diploma earlier.

I declare that I have faithfully acknowledged and referred to the works of other researchers wherever their published works have been cited in the dissertation. I further certify that I have not willfully taken other's work, para, text, data, results, tables, figures etc. reported in the journals, books, magazines, reports, dissertations, theses, etc., or available at web-sites without their permission, and have not included those in this M.Tech dissertation citing as my own work.

Date: 08 / 08 / 2020



Signature
Anum Kamal
Enroll. No. : 1400100477

COPYRIGHT TRANSFER CERTIFICATE

Title of the Dissertation: **Path Optimization: A Potential Approach to Path Planning for Vehicle Navigation**

Candidate Name: **Anum Kamal**

ANUM KAMAL

ACKNOWLEDGEMENT

I am highly grateful to the Head of Department of Computer Science and Engineering for giving me proper guidance and advice and facility for the successful completion of my dissertation.

It gives me a great pleasure to express my deep sense of gratitude and indebtedness to my guide **Dr. Faiyaz Ahmad, Assistant Professor, Department of Computer Science and Engineering,** for his valuable support and encouraging mentality throughout the project. I am highly obliged to him for providing me this opportunity to carry out the ideas and work during my project period and helping me to gain the successful completion of my Project.

I am also highly obliged to the Head of department, **Dr. Mohammadi Akheela Khanum (Associate Professor, Department Of Computer Science and Engineering)** and PG Program Coordinator **Dr. Faiyaz Ahamad, Assistant Professor, Department of Computer Science and Engineering,** for providing me all the facilities in all activities and for his support and valuable encouragement throughout my project.

My special thanks are going to all of the faculties for encouraging me constantly to work hard in this project. I pay my respect and love to my parents and all other family members and friends for their help and encouragement throughout this course of project work.

Date: 08 / 08 / 2020

Place: Lucknow

INDEX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence
UAV	Unmanned Aerial Vehicle
IS-IS	Intermediate system to intermediate system
GA	Genetic Algorithm
GPS	Global positioning system
DP	Dynamic Programming
PPCR	Planning of Coverage region algorithm
PSO	Particle Swarm Optimization
IPSO	Immune Particle Swarm Optimization
GUI	Graphical User Interface
TSP	Travelling salesman problem
ACO	The Ant colony Optimization
MATLAB	Matrix laboratory
TCS	Time - constrained Heuristics search
DoS	Denial of Service
DDoS	Distributed Denial of Service
CPU	Central Processing Unit
ICMP	Internet Control Message Protocol
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service

ABSTRACT

An efficient path optimization is said to find the shortest path from source to destination that takes less time, avoiding heavy traffic, where time being considered as the most important factor. But such is not always the situation. There has been a lot of research done on finding the optimized routes for many purposes like movement of autonomous vehicles, robot movement, aircraft movement, vehicle navigation etc. All these past researches focuses on finding a single efficient path based on a single predefined factor such as time or energy consumption. Most of the past approaches of path optimization had used algorithms like dijkstra, A* and Genetic algorithm.

In this dissertation, work has been done to improve the ability of already existing researches on path optimization. It aimed at developing a more efficient path optimization algorithm that helped to overcome the major drawbacks of the previous system. The objective was not only to focus on the three factors namely time, distance and traffic but to include availability of resources / locations along the path to visit according to the customer specification and it also helps in energy consumption. Another advantage of this research is that it also focused on finding out multiple routes from the source to destination, giving a user various path options to decide from, according to the current need.

The combination of algorithms: Floyd - Warshall and 0/1 Knapsack were used to get the optimal paths. While the best path according to the customer need was found using LSTM Algorithm based on customers feedback and past experiences. All the above finding were presented on a map for better understanding. The major advantage of this approach is its potential to navigate routes according to the specified resources, simultaneously providing multiple route options to choose from.

CHAPTER - 1

INTRODUCTION

1.1 PATH PLANNING AND OPTIMIZATION:

Path planning is also called Motion planning. It is a computational problem that can be solved automatically by the computer, following a sequence of a valid sequence of constraints which allows movement of the objects from its source to its destination. For example, consider a navigating car in a city to a distant waypoint. It should find a path while avoiding footpaths, buildings, unconstructed areas etc. A path planning algorithm would take all these description and more as input, before providing instructions of a path suitable for movement of the car within a city to reach its destination

It can be divided into two main categories; (i) local path planning and (ii) global path planning. In local path planning, the computational device is well aware about the area to be navigated as it is very limited and thus the destinations can be reached by following some predefined paths. Whereas, in global path planning as the name says the area is not limited and the computational device does not have the complete knowledge required to freely navigate the area [33]. Thus global path planning has a very few application because of the lack of knowledge of the area as well as less robustness, but in the local path planning, the methods used, shows more flexibility and are capable of providing optimal path both in partially known/unknown areas [34].

Hence, in short, path planning refers to finding the shortest route between two points i.e. source and destination. It starts from the source and repeatedly search for the path until it reaches the end point [15]. However finding an optimal path complicates the situation. It has now become a primary concern of finding the shortest feasible path [1]. There is a huge vast difference between finding a path and an optimal path. There are two primary problems in path finding (i) finding a path between two nodes (ii) find an optimal shortest distance between two nodes [18]. Finding an optimal and shortest path is the core of path optimization [11].

Thus the path optimization can be referred to as finding an alternative path which is most cost, time and distance effective, which is able to give the maximum attainable performance under the stated constraints, by focusing more on the desired factors and less on the undesired ones. The basic factors that influence the shortest path are mentioned in the following section.

1.2 FACTORS AFFECTING PATH OPTIMIZATION:

There are various factors the affects the path planning and optimization. Figure 1.1 shows the classification of the factors involved in path optimization. All the factors are classified into four main categories:

1.2.1 BASIC

The most common factors that are the base of any path optimization algorithm are the basic factors. They are:

- ❖ **Distance:** The distance between the source and the destination matters a lot. The longer distance generally results in longer time.
- ❖ **Time:** It is the most significant factor in optimization. No matter if the distance is long but if the time taken to reach the final destination is less the path is said to be an efficient one. The time also varies on the area and vehicle dynamics such as the obstacles in the path, traffic, the friction between tires and the road, the vehicular speed etc.
- ❖ **Cost:** Another common factor that affects optimization is the cost or total expenditure. It mostly varies with the cost of fuel or battery used.

1.2.2 ENERGY CONSUMPTION

Another important category is the consumption of energy while travelling. Energy can be in two forms namely Fuel and battery. Both forms of energy serves the main purpose that is to run a machine like in cars, busses, trucks, the form of energy utilized is fuel while in autonomous vehicles, robots or even games the energy source is the battery. Usually it is stated the more the friction between the tires or the obstacles in the way like in complex environment, more will the energy loss. It helps deciding whether the particular machine, be it a robot or vehicle, is capable of taking the specified route or it will stop even before reaching the destination or finishing its task.

1.2.3 SAFETY CONCERNS

While planning an optimal route some safety concerns are also taken into consideration. They are:

- **Drivers Safety:** In the situation where a person wants to reach a certain destination as fast as it can the speeding not only put the Driver but also the passenger’s life in danger.
- **Road Safety:** The safety in the path travelled is also of concern.
- **Vehicle Safety:** Driving the vehicle or speeding a machine beyond its safety limits could also prove to be very hazardous.

1.2.4 AESTHETIC

This category summarizes factors like scenery, noise and smell. These factors can also at some point influence the path optimization experience.

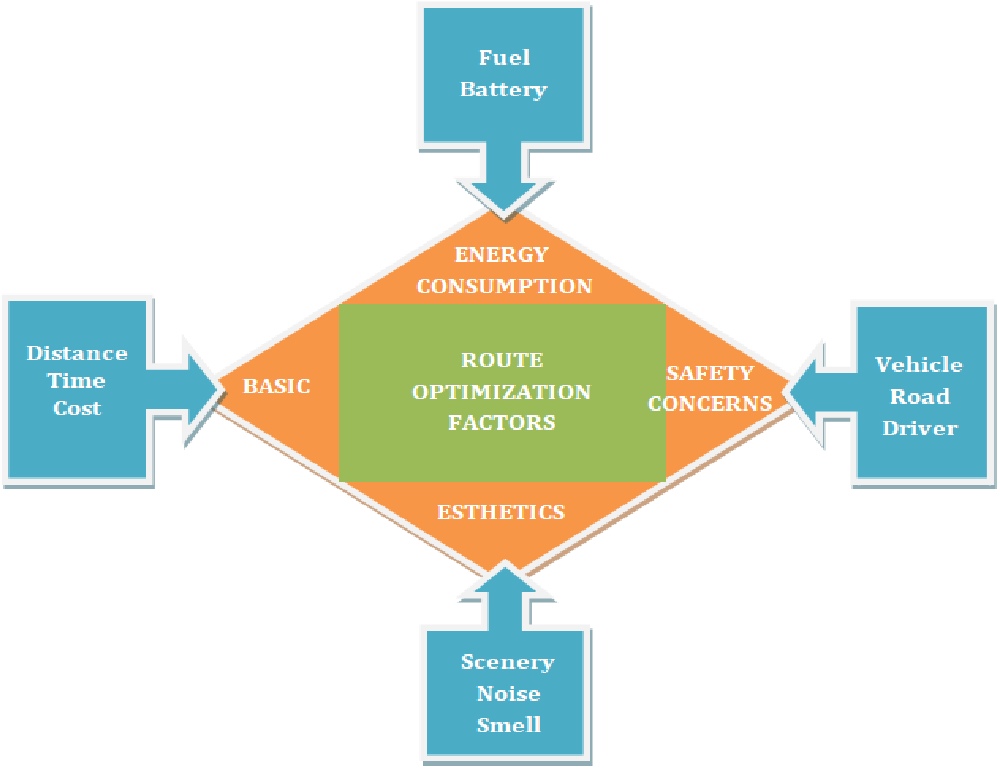


Figure 1.1: Factors affecting path optimization.

1.3 NEED FOR PATH PLANNING AND OPTIMIZATION:

The advancement in travelling technologies has drastically reduced a man's effort for path planning. The fast pacing economical world and rapidly changing working conditions has reduced the time on people's hand, thus, wasting it on activities like effective path planning and transportation is futile.

Also in factories due to increase in production in order to meet the requirements of customers, the manual efforts of the employees is not sufficient to fulfill the needs. Thus here also the need of computational devices arises, that can navigate through a factory and perform various tasks in place of man.

Similarly, there are various other areas where the need of finding the feasible path for the movement of a computational device is necessary.

Thus path planning and optimization plays the major role in various other domains such as vehicle routing, which is movement of the vehicle within a city, Artificial intelligence, which is mostly used in gaming for the movement of various characters, logistics movement, for the movement of automated vehicles, emergency evacuation, it is important during the time of disaster or an accident, delivery drones or autonomous drone, for finding a correct path to reach the final point safely and many others [19].

All these problems have encouraged many researchers to focus on effective path planning and optimization techniques.

1.4 VARIOUS APPLICATIONS OF PATH PLANNING AND OPTIMIZATION:

Optimal Path finding plays vital roles in many domains like vehicle routing, robot path planning, logistics movement, Artificial Intelligence (AI), games, emergency evacuation and various others [19]. In terms of vehicle route optimization it is necessary to have a shortest path following all factors of optimal solution. Looking for a shortest route between two locations in the road network is a very challenging task in vehicle routing, distribution and logistics industry [32]. And in many transportation application that involve real road network choosing suitable and efficient routing algorithm also plays a significant role [20]. Figure 1.2 shows inputs, output and updation in the process of vehicle routing.

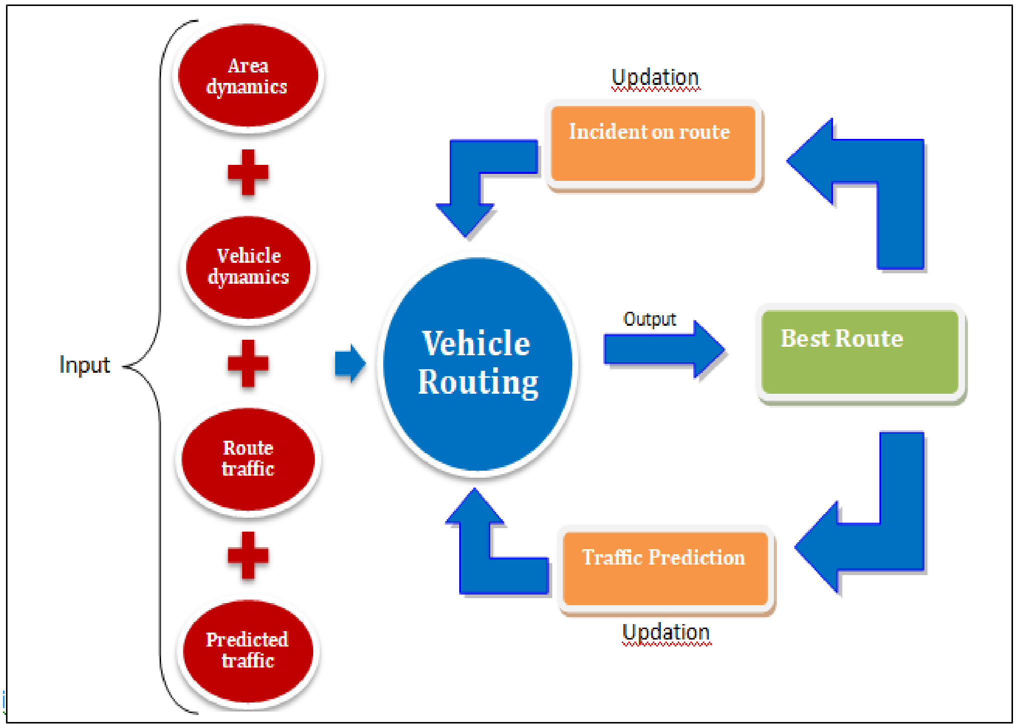


Figure 1.2: Basic steps involved and the inputs, output and updation in the process of vehicle routing. [Source: 32]

Similarly in games that require path finding and optimization to make it more human like [1]. According to M. Cowley [21], the determination of how close a robot can get to "human intelligence" depends upon its similarity to human and his behavior. Thus path finding plays a very important role in commercial games and robot navigation. It has become a core component in the movement system of game AI [16]. In this it concerns in a way in which an object or a game character is able to find its path omitting the obstacles, because of this, many AI based platforms and tools are being developed [15]. Similarly in robot navigation the planned path should be optimal and collision-free and should be capable of achieving a certain desired goal in a complex environment [5].

In events like emergency evacuation, movement of UAV and emergency planes and vehicles where there are cases of accidents, disaster, earthquakes or any other calamity, the need to provide an optimal and shortest path in minimal time also increases [8]. In such situation any mistakes could cause severe damage to life and property.

Because of these reasons and applications researches have been done and are still going on finding an algorithm capable of providing such optimality.

1.5 RELATED WORK

There has been many researches in the past and are still going on intensively in the field of optimized energy efficient path planning for various purposes like vehicle routing, disaster management, routing of autonomous vehicles, robot movement in games and otherwise, UAV movement etc. All these researches show various algorithms and techniques for an efficient path planning.

1.6 REVIEW OF ALGORITHM USED IN PATH PLANNING AND OPTIMIZATION

Figure 1.7, shows basic classification of algorithms in terms of path planning and optimization.

The most frequently used algorithms in a majority of researches were:

1.6.1 DIJKSTRA ALGORITHM:

Dijkstra algorithm is one that is capable of finding the shortest paths between nodes in the graphs. It was introduced by a computer scientist Edsger W. Dijkstra in the year 1956, which after three years was published. [22][23] It was stated as the best algorithm capable of finding a shortest path from a particular node to any other nodes in the graph [24]. Being one of the most extensively used algorithm it basic idea is to identify a shortest distance from the source point to other destination points followed by using the paths lengths to find the shortest path iteratively.

It uses labels which are represented as ordered positive integer numbers or real numbers to label the nodes constantly The value of the label varies with the nodes . In order to store and query partial solution which are sorted by distance Dijkstra Algorithm uses data structure such as min - priority queue. There a various variants od dijkstra algorithm based on different factors like different data structure or for obtaining different results.

The various applications of this algorithm is in finding the shortest route from one city to another, in network routing protocol, in IS-IS (intermediate system to intermediate systems and in various other applications.

Apart from being the more popular and commonly used algorithm it also has some deficiencies. In case of complex areas for example the Urban area the speed of the traditional dijksta decreases drastically. Whereas in terms of data it requires a vast, large amount of data space in order to store the data of its finding.

The worst case performance of this algorithm is:

$$\boldsymbol{O}(|\boldsymbol{E}| + |\boldsymbol{V}| \log |\boldsymbol{V}|)$$

where,

|E| is the number of edges

|V| is the number of nodes

Figure 1.3, illustrates an example of dijkstra algorithm, where it shws how a shortest path is found between the source and destination using dijkstra algorithm.

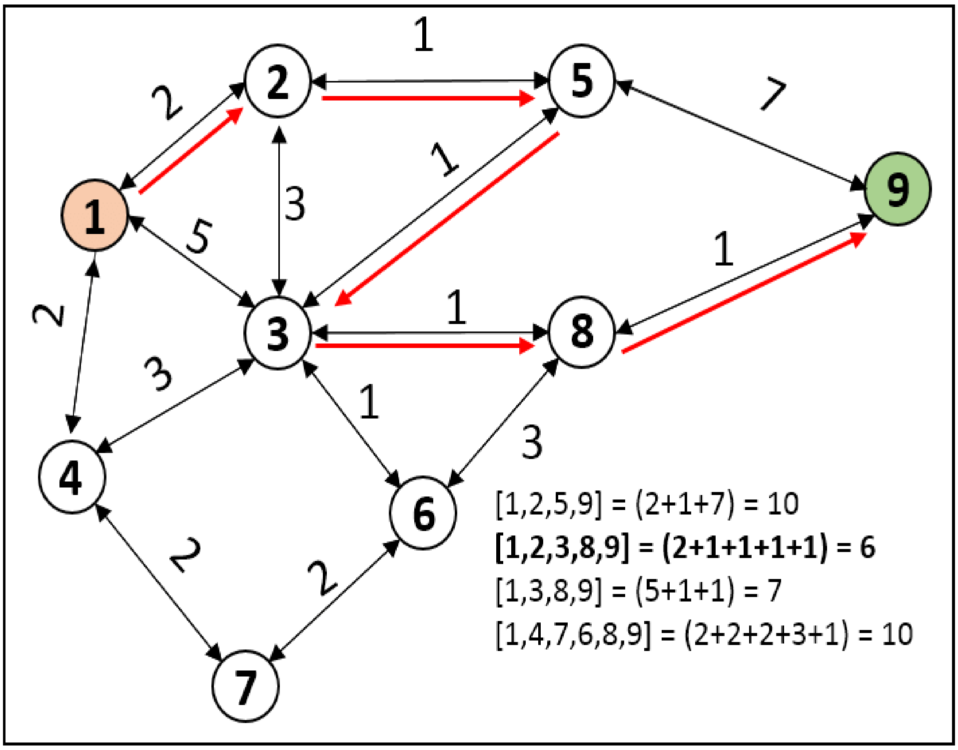


Figure 1.3: An example illustrating the use of dijkstra algorithm.

1.6.2 A* ALGORITHM

In the year 1967 Peter Hart, Nils Nilsson and Bertram Raphael proposed an informed search algorithm called the A* search algorithm [25]. It is extensively used in the fields like computer science and pathfinding for problem solving [1]. The creation of A* algorithm was a part of “the shakey project” having the aim of building a mobile robot capable of planning its own actions.

It is also a breath first search algorithm as it is devised in terms of weighted graphs. It starts from a particular node and aims at finding that path from the source to destination which have minimum distance, shortest time and smallest cost. Being seen as n extension of Edsger Dijkstra’s 1959 algorithm A* is capable of achieving better performance by using heuristics as its guide in search.

What makes this algorithm more special is that it has “brains”. This means that it is extremely smart algorithm and that’s what makes it unique. A* operates by maintaining two list namely: (i) open list which is initialized by the start node and contains all the unvisited nodes and (ii) close list which begins with an empty node and comprises of all the visited nodes [1].

At the beginning of each loop A* determines which paths to take, on the basis of the cost of the path and an overall estimate of the cost needed to extend the path upto the destination. A* selects the path that minimizes

$$\boldsymbol{f(n) = g(n) + h(n)}$$

Where,

n is the next node,

G(n) is the cost of the path from start to n,

and h(n) is a heuristic function.

The heuristic function is problem specific and its major goal is to estimate the cost of the cheapest path from the source starting place to the goal. There can be basic three approximation heuristics, namely: Manhattan Distance, Euclidean Distance and Diagonal Distance For example, when we search for the shortest route on a map, h(x) may represent a straight-line distance to the destination, because it is physically the smallest distance possible between any two points. Figure 1.4, shows an example of straight line distance when Euclidean distance is used as heuristics in A* algorithm

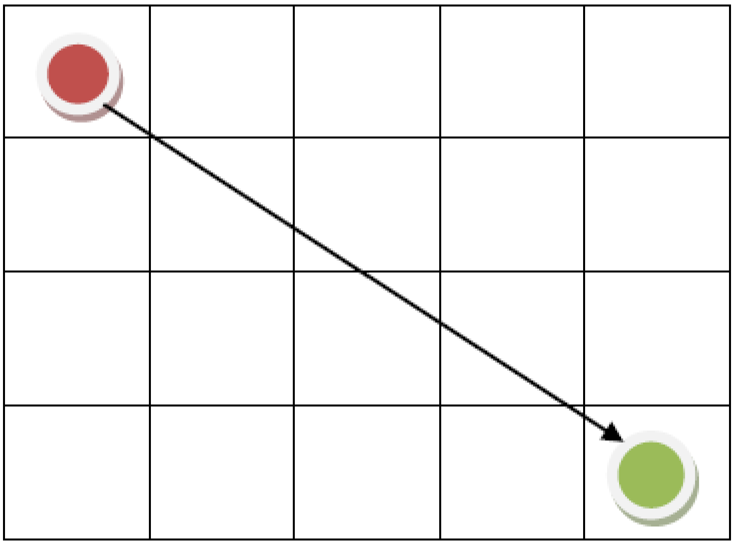


Figure 1.4: An example of straight line distance when Euclidean distance is used as heuristics in A* algorithm

In the recent years A* search algorithm has become a very popular algorithm. It is a technique that is very commonly adopted for path finding and graph traversal. Many researches have even stated that it gives faster results than Dijkstra [26]. The main reason behind this is that like dijkstra it does not need to repeatedly visit each node in order to provide an optimal path [1]. This algorithm provides characteristics like completeness, optimality and optimal efficiency.

The major drawback of it is space complexity. In terms of execution time, where the number of nodes in the map increases, A* becomes a very expensive approach [27]. With the larger map size A8 requires more memory before it can process any solutions [28]. This space limitation is the main cause that lead to the introduction of several other variants of A* algorithm. Originally, A* was made as a general graph traversal algorithms but now, the basic purpose in terms of pathfinding is video games [29]. Various other applications are solving diverse problems, solving the problem of parsing using stochastic grammers in NLP [30], also an informational search with online learning [31].

1.6.3 GENETIC ALGORITHM

Genetic algorithm is a metaheuristic method which follows the process of natural selection. It was introduced in 1960 by John Holland and was based on the concept of Darwin's theory of Evolution, which was further extended by his student David. E. Goldberg in 1989 [36].

It is an adaptive and global probability algorithm which follows biological genetics and evolutionary process in natural environment, to solve a problem [17]. This robust technique belongs to the class of evolutionary algorithms such as, simulated annealing, threshold acceptance and some branch and bound [35]. Unlike most stochastic search methods, which work on single problem, Genetic algorithm works on a population of solutions [13].

GA creates a population of genomes on which biologically inspired operators are applied such as mutation, crossover and selection [37]. Figure 1.5, shows the basic process of genetic algorithm. In the process it uses various selection criteria to pick the best the best individuals which are called population [13]. Then with the help use of variety of genetic operators (such as mutation, crossover and selection) it produces the next generation. This is continued until the terminal condition is met.

The process of genetic algorithm is explained in detail in the flowchart provided in figure 1.6.

The genetic algorithm is very simple and is used to produce high quality solutions to optimization and search problems. Its parameters can be modified based on the requirements to suit the purpose.

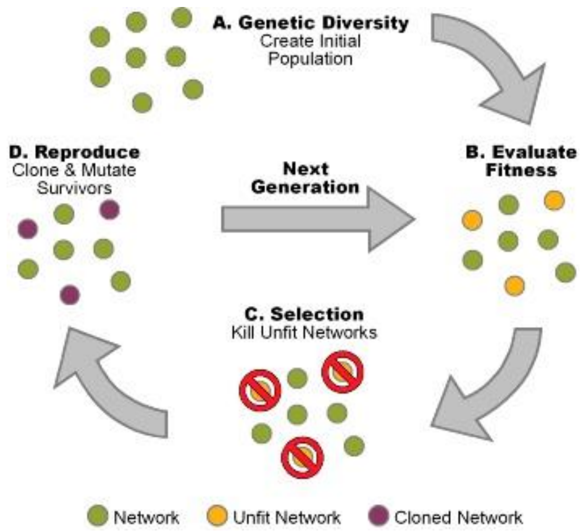


Figure 1.5: Genetic algorithm process

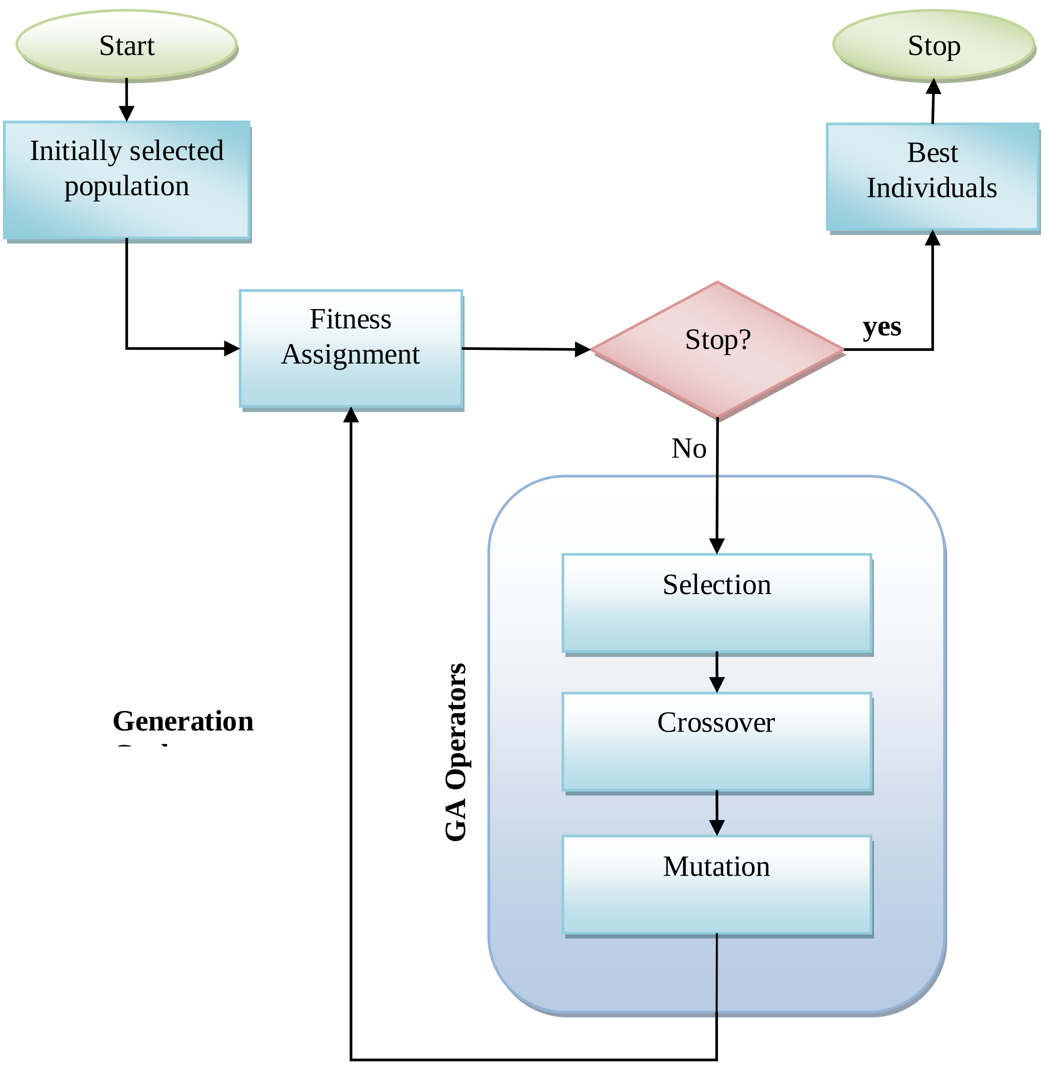


Figure 1.6: Flowchart showing detailed process of genetic algorithm

1.6.3.1 NEED FOR GENETIC ALGORITHM

- Easy to understand.
- GA operates on a cluster of points, not just a single point.
- GA supports multi-objective optimization.
- GA does not use deterministic rules but instead it uses probabilistic transition rules.
- GA is good for complex environment environments.
- It can be easily parallelised.
- GA can operate on mixed discrete/continuous problem.

1.6.3.2 LIMITATIONS OF GENETIC ALGORITHM

- It requires an expensive fitness function evaluations for optimal solutions to complex high-dimensional problems.
- The stop criterion is not always clear in every problem.
- Operations on dynamic data set is difficult
- It is computationally expensive

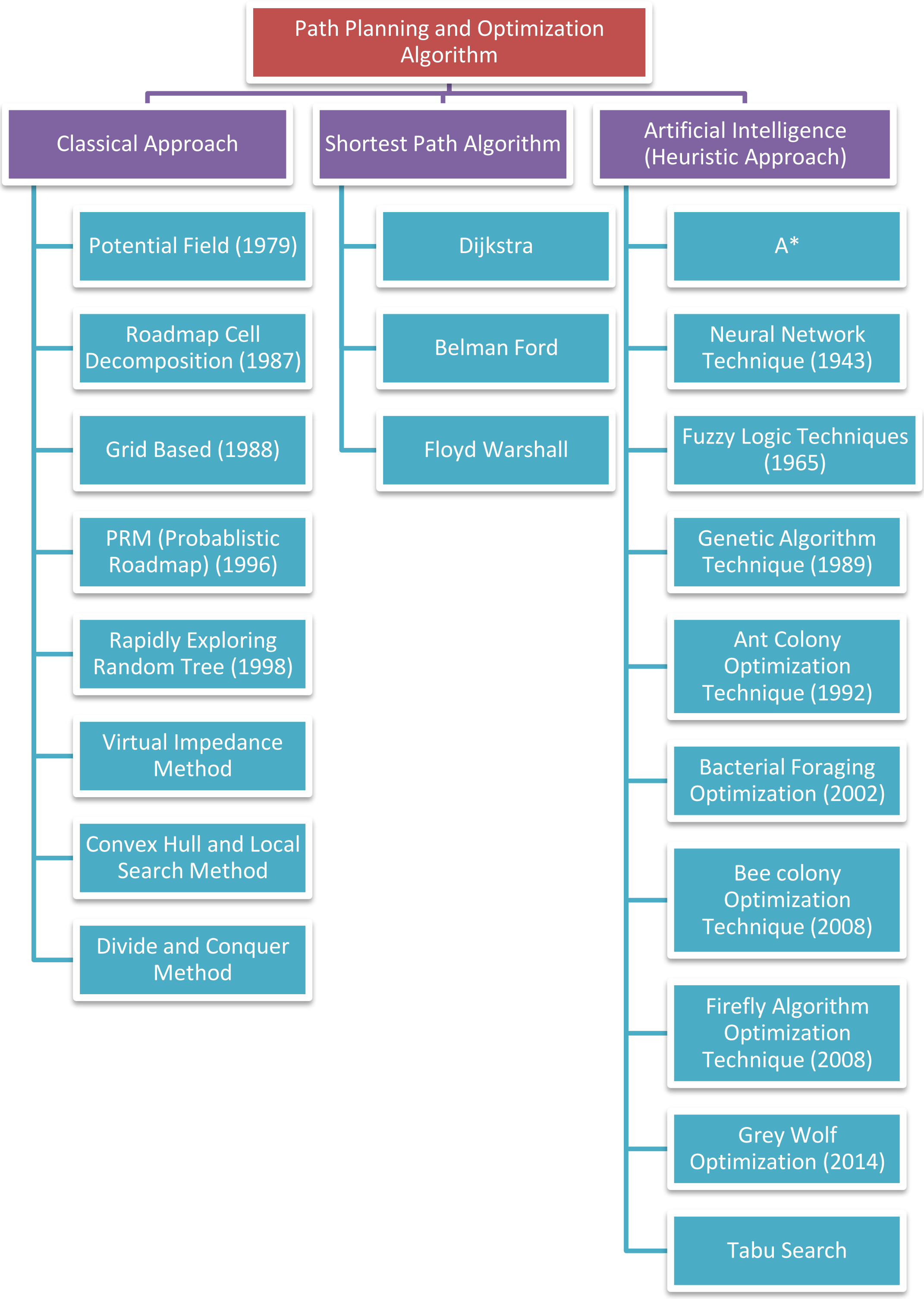


Figure 1.7: Classification of various Path Planning and Optimization Algorithms

1.7 DISSERTATION OUTLINE

The organization of rest of the dissertation is as follows:

Chapter 2

In this chapter, security challenges and issues path planning and optimization approach is discussed. Objectives of this chapter are to explain different types of attacks on cloud and GPS in detail.

Chapter 3

In this chapter, there are reviews from various national and international journals and publications. It is done to identify the real problem statement for doing appropriate research.

Chapter 4

In this chapter, the proposed work is discussed and explained in detail for finding optimal path on the specific parameters and factors.

Chapter 5

In this chapter the metrics that were utilized to measure the performance of proposed work along with diagrams that illustrate the performance measurements, the implementation details and results of the detection mechanism is discussed. In order to give more clear view of the implementation details involved part of the code and results are presented as the algorithm, flow chart and graphs.

Chapter 6

In this chapter conclusion and some of the future scopes is discussed of this work.

CHAPTER – 2

SECURITY BACKGROUD

There are various security issues that occurs when creating a project that involves large amount of data processing, various different parameters and constraints, use of networks such as GPS, user authentication etc.

Some of the security issues that can occur in the proposed work are as follows:

2.1 DENIAL OF SERVICES

Denial-of-Service attack(Figure 2.1) is a cyber attack intended to hamper the normal functioning or completely shutting of a machine, device or a network preventing its intended users (i.e. members, office employees, account holder etc.) to access it. The core purpose of this attack is to overload the target with the redundant requests, send in abundance, or send certain information that triggers a crash, which cause the system to shut down completely depriving all the requests which were legitimate from being completed [38].

This attacks targets influential and famous organization of banking, commerce, media, or government. Denial of service attack is not mainly used for the purpose of theft or hampering significant information but for destroying of targeted network or machine, which as a result can cost a victim lot of time and money to repair.

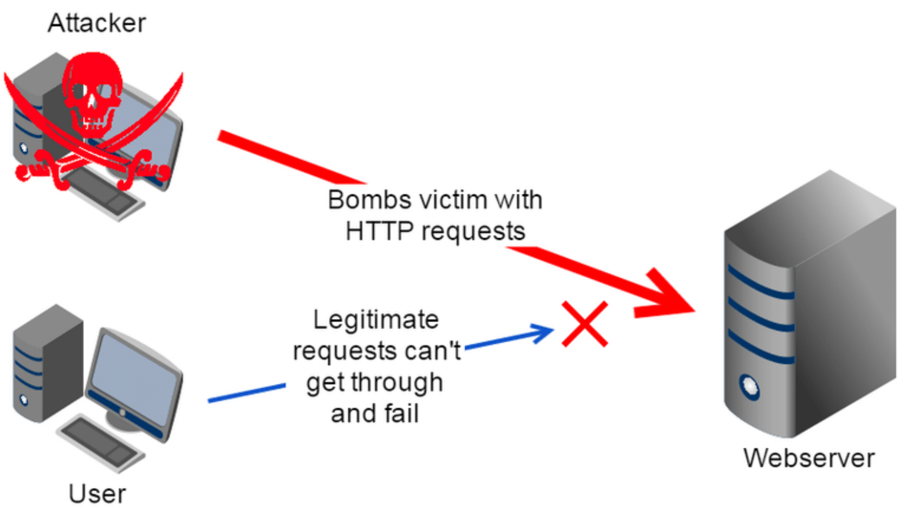


Figure 2.1: Denial – of – Service attack

2.1.1 HOW DOES THE DENIAL OF SERVICE ATTACK (DoS) WORKS?

As mentioned previously in this chapter the denial of service attack is successful only if it is capable of overflowing the targeted system or network, denying the other authorized users to access it. This attack can not only hamper the work of the users but can also severely damage the system resources.

This attack targets either the system physical resources or it's network connections. When targeting physical resources it sends a huge quantity of redundant requests to the server making it unable to handle all of them at once. This prevents it's users to us the resources and the system to crash.

Whereas, when targeting the system's network connections, it sends invalid packets to the network in a huge amount. This overwhelms the network, severing the connection with the system/s.

Figure 2.2, shows the basic working of the DoS attack

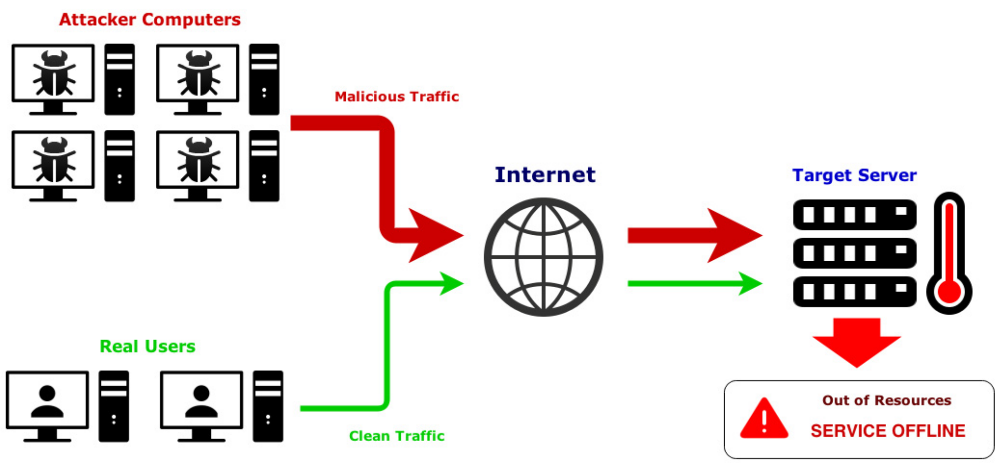


Figure 2.2: Working of Denial – of – Service attack

2.1.2 TYPES OF DENIAL OF SERVICE (DoS)

Denial-of-Service attacks can be categorized in 2 forms, where on crashes the service or a system while the other floods a network. They are:

❖ **Buffer overflow attacks:**

In order to consume all the disk space, memory and CPU time available, this attack overflows the buffer of the targeted system, causing the system to hang, slow down or show other unfavorable behavior, thus resulting in denial of service (Figure 2.3).

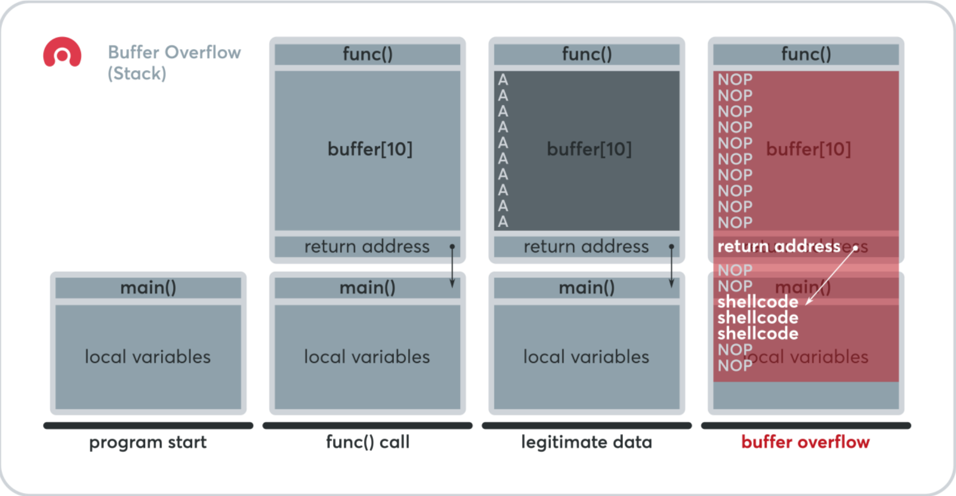


Figure 2.3: Buffer overflow (DoS) attack

❖ **Flood attack:**

In this form of attack the denial of service happens when the targeted server capacity is overflowed with the abundant amount of packets send by some malicious source or person (Figure 2.4).

All kinds of denial of service attacks falls under the above mentioned categories the most serious attacks being the distributed once [39].

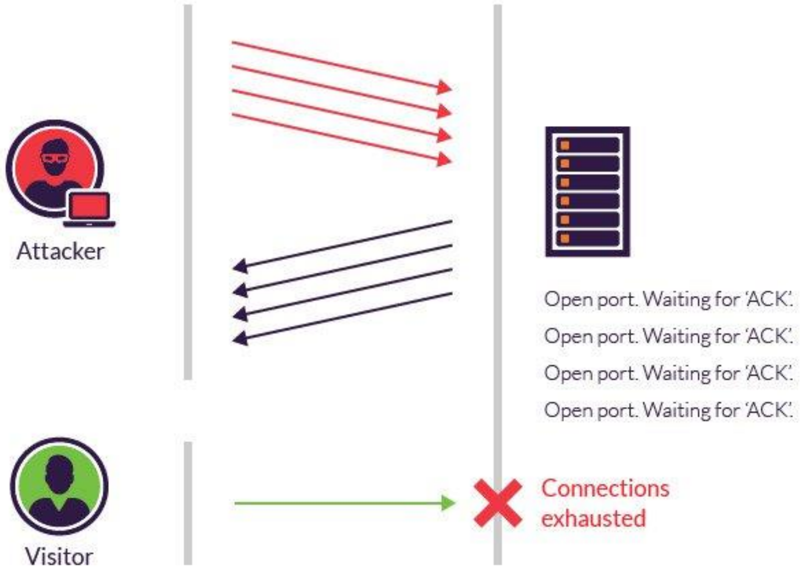


Figure 2.4: Flood (DoS) attack

2.1.3 DISTRIBUTED DENIAL OF SERVICE (DDoS) ATTACK:

Distributed denial of service attack (Figure 2.5) is the most serious, large-scale DoS attack in which the attacker uses multiple source i.e. the different IP addresses or machine to attack the target at the same time. This is done from thousands of malware infected hosts [40]. What makes this attack a very serious one that the redundant requests or packets comes from various sources at the same time, making it hard for the victim to identify the host. Due to the distributed nature of the attack, it is neither possible to stop this attack as it cannot be resolved by any simple filtering techniques, nor the authorized users can be distinguished from the non-authorized ones, because there is no sole point of origin.

This type of DDoS attacks have increased considerably over the years and by 2016 rate of attacks exceeded by terabit per second [41][42]. There various types of DDoS attacks, but some common examples of these attacks are, UDP flooding, SYN flooding and DNS amplification [43][44].

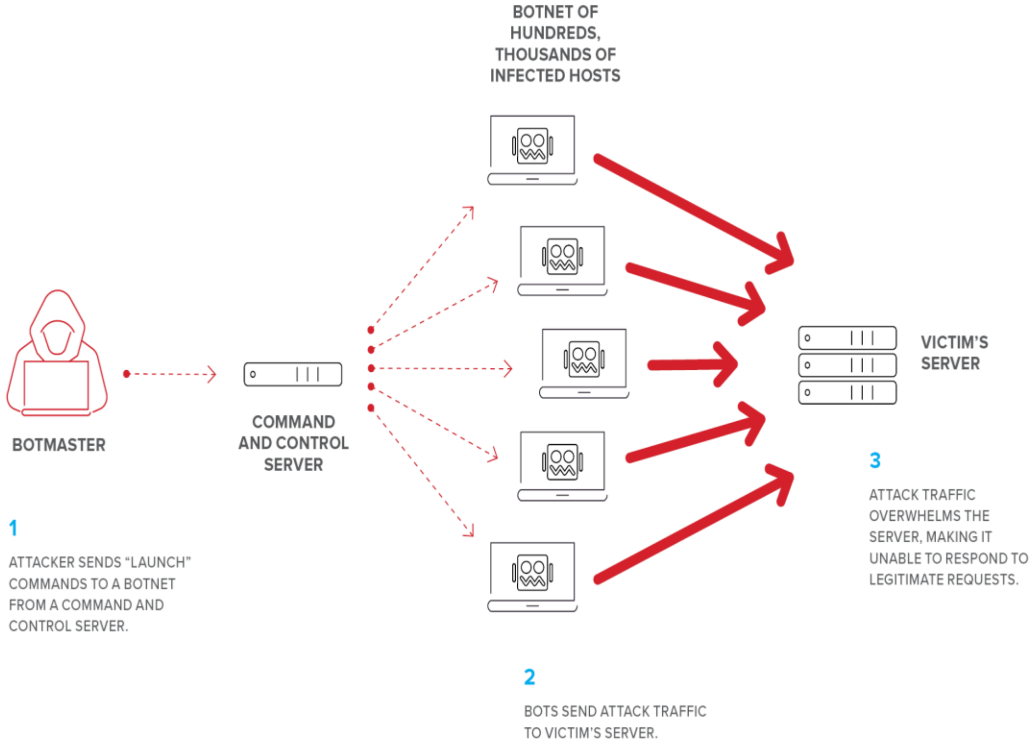


Figure 2.5: Distributed denial of service (DDoS) attack

2.1.4 DIFFERENCE BETWEEN DoS AND DDoS

The basic difference between DoS and DDoS is the number of malicious host involved in the attack.

As is DoS, the attack is from a single malicious host which overwhelms a particular system or network, preventing it to process the request of other authorized users, misbehavior of the system and even crashing of the system.

Whereas in DDoS, the attack is not from one source only but from multiple malicious hosts, onto the targeted system, at the same time. This attack is impossible to stop as there are thousands of malicious hosts involved, which are distributed all over the network, also it is very difficult to distinguish between the authorized users and malicious hosts. Hence, this is the most dangerous and commonly used from of DoS attack (Figure 2.6).

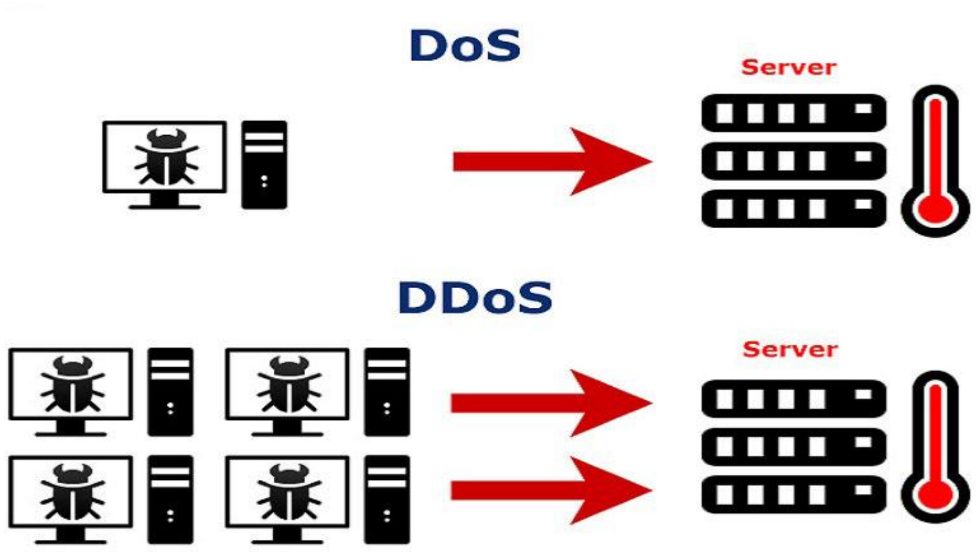


Figure 2.6: Difference between DoS and DDoS attacks.

2.1.5 SOME HISTORICALLY SIGNIFICANT DoS ATTACKS

In the past, the DoS attacks utilized the vulnerabilities of the targeted system or network, software and hardware. There use have been reduced recently as now they are usually made of distributed nature, as they are capable of causing more serious disruptions and are comparatively easy to create than the traditional DoS attacks.

Even though the DDoS are more dangerous, still some dangerous DoS attacks are also recorded in history. They are:

- **Smurf attack**: In this the broadcast address of a vulnerable network is utilized by the attacker to send some burlesqued packets, which results in flooding of the targeted IP.
- **Ping flood**: By overwhelming a target with more pings or ICMP packets, than it can efficiently respond to this attack causes denial of service.
- **Ping of Death**: It is somewhat similar to the ping flood attack. But here a malformed packet is send to the target, which crashes the system entirely.

2.1.6 HOW CAN ONE TELL IF A SYSTEM/NETWORK IS UNDER DoS ATTACK ?

The are various indications that can tell whether a particular system is under denial of service attack or not. They are:

- The files will take longer time to load, due to the slow network performance.
- Sudden loss of connection to a network.
- Unresponsiveness of a particular website, when loaded on web page.
- Increase in processing time.
- Unnatural behavior of the system.

In the proposed work this attack is can be prevented because of the following reasons:

- Each client using the system has to be registered using his/her personal information. When registered the client is provided with the secret key through which he/she can access the system. Hence it cannot be accessed by any other unauthorized user.
- The secret key provided to the user is used in accessing the location of the user using Global positioning system (GPS). This key can be activated and deactivated by the user as and when required.

2.2 CLOUD SECURITY

Cloud security is a type of Cyber Security, also called as cloud computing security. It has various set of policies, controls, procedures and technology that are used to secure cloud based systems, data and environment from both external and internal cyber security threats such as theft, leakage and destruction.

The cloud security can be optimized according to the business need, from authentication to filtering of traffic. Since the management and optimization of these rules can be can be done from a single place, within the organization or even outside, overheads created by its administration can be reduced drastically.

2.2.1 CATEGORIES OF CLOUD COMPUTING

On the basis of category of cloud computing used, the cloud security also differs. Following are the four chief categories of cloud computing (Figure 2.11), according to which cloud security also can be categorized:

2.2.1.1 Public cloud

It includes all three kinds of infrastructures such as software-as-a-service (SaaS), infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS). It is called a public cloud as it can be used all and is operated by a public cloud provider (Figure 2.7).

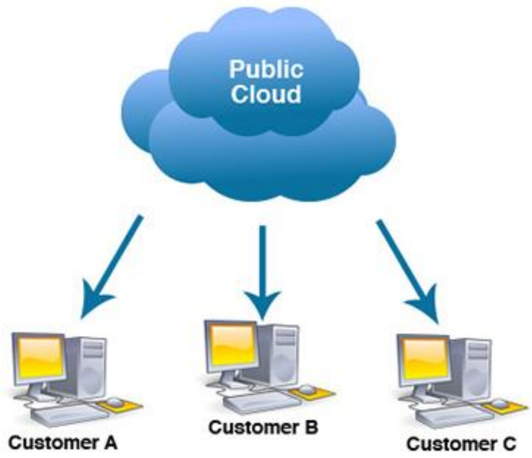


Figure 2.7: Public cloud

2.2.1.2 Private cloud

This type of cloud is used by a single organization. It can be operated by both i.e. the internal staff of the organization or any external third party public cloud provider (Figure 2.8).

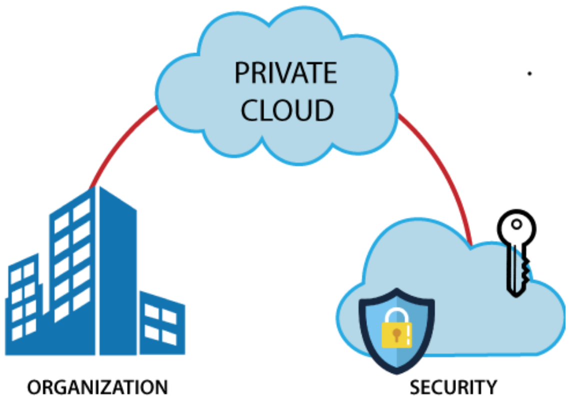


Figure 2.8: Private cloud

2.2.1.3 Hybrid cloud:

This cloud combines the services of both public as well as private cloud. It can also be handled internally as well as externally (Figure 2.9).

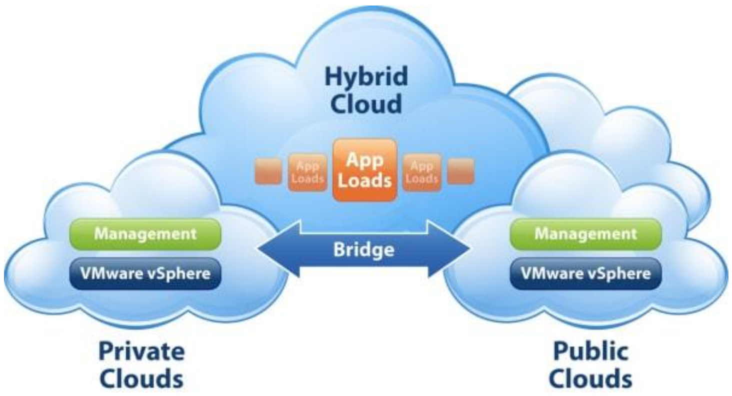


Figure 2.9: Hybrid cloud

2.2.1.4 Community cloud:

This type of cloud is shared among various small organizations and can be managed either by any internal person from one of the organization or by any third party (Figure 2.10).

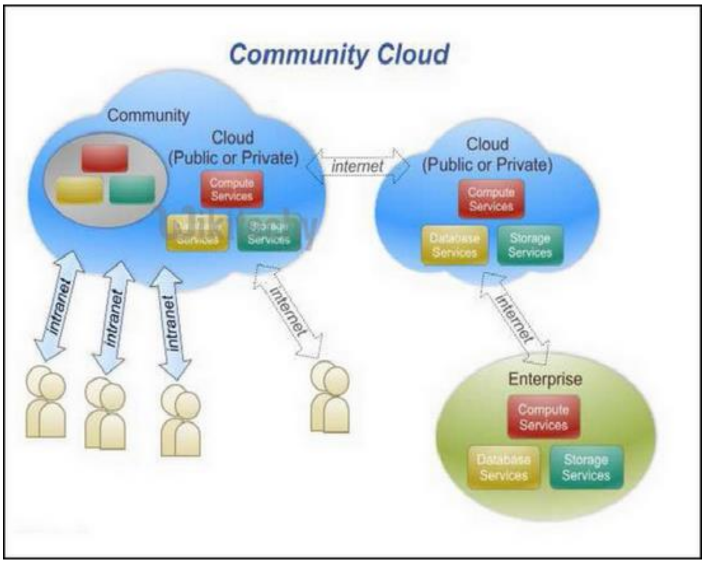


Figure 2.10: Community Cloud

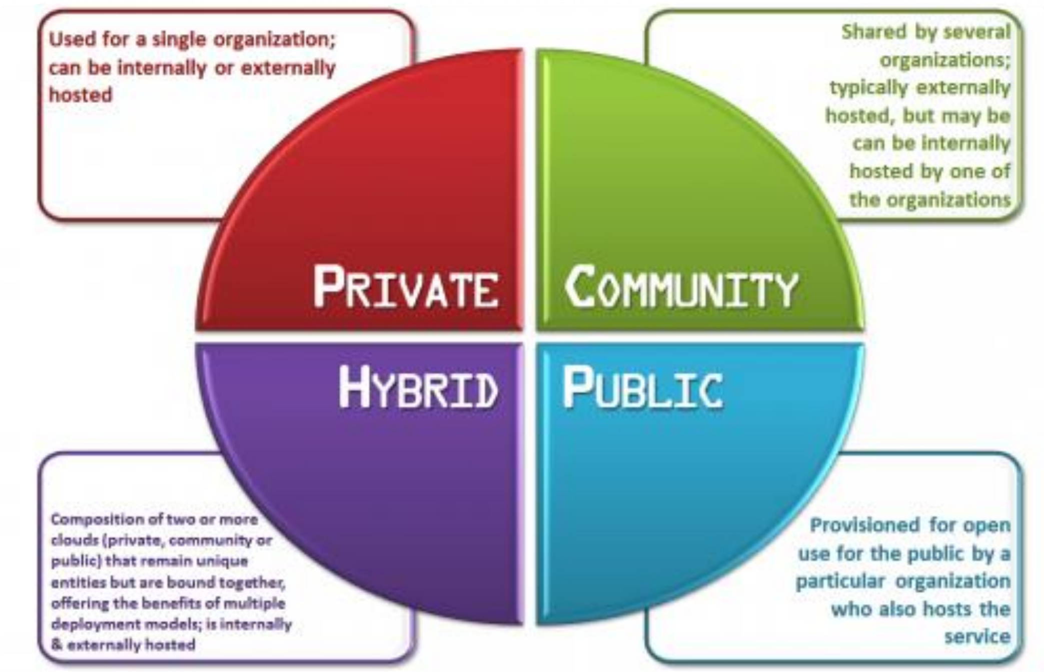


Figure 2.11: Types of cloud

The major security issues that arise are in public cloud as it can be accessed by anyone. As the security of this cloud is handled by the third party or an external cloud service provider, who attempt to avoid security issues from the services which they provide, they cannot control the customers using the services. They also have no control on the data added by the customer to the cloud or who is accessing the cloud, which can in some way weaken the cloud security.

Thus in public cloud service type, there are different set of responsibilities for both i.e. the customers and the service providers, which is distributed according to the service type as follows:

- **Software-as-a service (SaaS)** – here security of data and user access are the sole responsibilities of the user.
- **Platform-as-a-service (PaaS)** – here the responsibility of the customer is also to secure application loaded on cloud along with the data and user access
- **Infrastructure-as-a-service (IaaS)** – here since the entire infrastructure in involved thus only securing the applications, users access and data is not enough, the protection of operating systems and network traffic is also necessary by the user.

2.2.2 WORKING OF THE CLOUD SECURITY

The basic need for cloud security is to have access control i.e. to be able to ensure that the data stored in the cloud can only be accessed by only legitimate users. In order to implement this cloud security provides a lot of tools and procedures (Figure 2.12). Some of the tools and procedures that the organizations used are:

- **Micro-segmentation:** Here the data center is divided into separate segments, so that they can be handled by an individual. This way the chances of having a security breech is minimized.
- **Next-generation firewalls:** The next generation firewalls comes with the application-aware technology to keep out advanced threats. It is more smart, efficient and effective.

- **Data encryption:** Keep data in encrypted form is also a way to prevent it from getting stolen or tampered, as in order to access it you need a particular key for decryption.

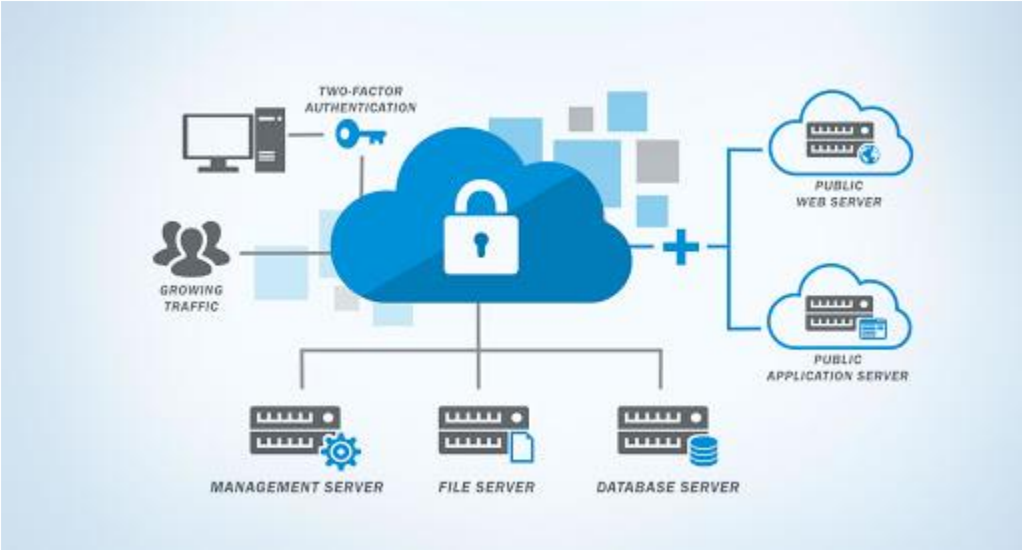


Figure 2.12: Technologies used in cloud security

2.2.3 BENEFITS OF CLOUD SECURITY

The cyber threats always attacks the data, in order to steal or tamper it. For this reason the cloud security comes into play as, cloud consists of large quantity of data from various user, making it a sole target for the attackers. Thus securing this data becomes a must. Various benefits that cloud security provides are:

- It provides centralized security to the cloud where the data is also centralized.
- The extra cost is also reduced. Due to the centralized security there is no need to pay for other dedicated hardware.
- All the security administration is done in a single place, thus manual configurations of security or security updates are not required
- It is reliable as with the right security measures, which are centralized, the users can use data and applications on cloud very safely.

- Various cloud security services provide live monitoring of data and hence improve its availability

As the technologies are growing more and more organizations can easily rely on cloud and cloud security. It allows them to easily use up resources along with security, in minimum cost.

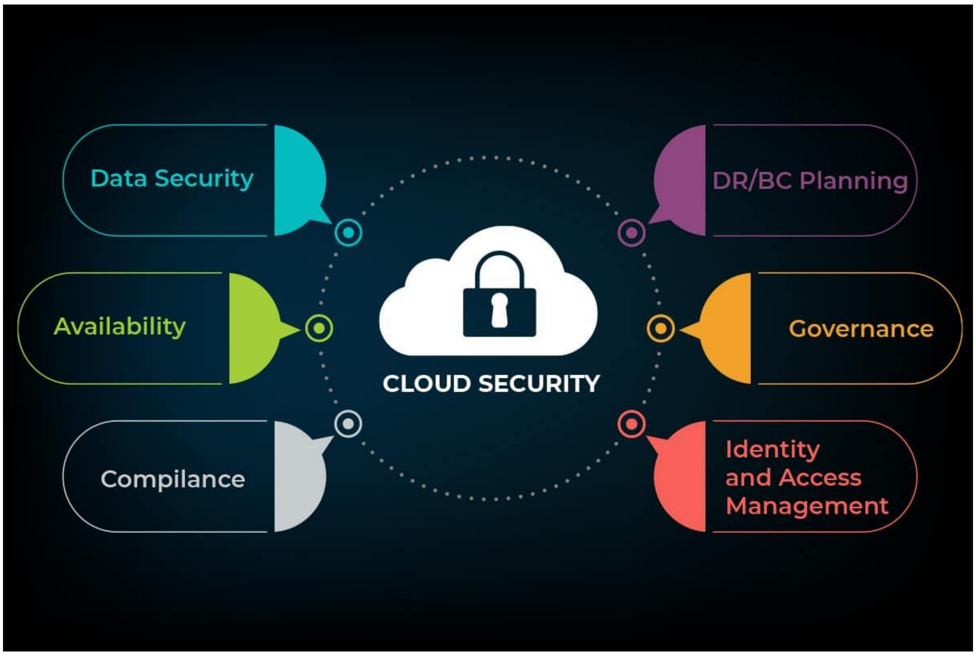


Figure 2.12: Benefits of cloud security

2.2.4 CLOUD SECURITY CHALLENGES

There are various security challenges when come to securing of public cloud as it is being managed by the third party and is used through internet. Some of these challenged are:

- The IT teams should have the capability of seeing into the cloud services itself in order to have the full control of data, since the cloud services on the public cloud is accessible to anyone or any device even outside the organization, the data is transparent.
- Where the third parties are involved for management of the cloud, IT teams of the organizations have very less control over the data on the cloud

- ❖ Due to free accessibility of the cloud data and applications, any user or device can access it and no traditional access controls works to restrict this data access.
- ❖ Another security threat is the cloud –native breech attack, where the data is stolen with the help of native functions of the cloud.
- ❖ Misconfiguration can turn out to be one of the major causes of cloud-native breech attack, which can allow an easy entry to the attacker. Then the attacker can easily expand, destroy and steal data. There is also a research showing that 99% of the times the misconfigurations get unnoticed by the user.
- ❖ There should always be a backup plan in case of attacks like cloud-breeches.There should be proper policies, procedures and tools for disaster recovery, which can allow the organizations to work even after the disaster.
- ❖ The threat is not only on the outside, a dishonest employee within the organization, who has access to the cloud services, can also be a cause of the security breech. According to one of the reports by McAfee Cloud Adoption and Risk, 85% of the organizations are suffering because of the insider threats.

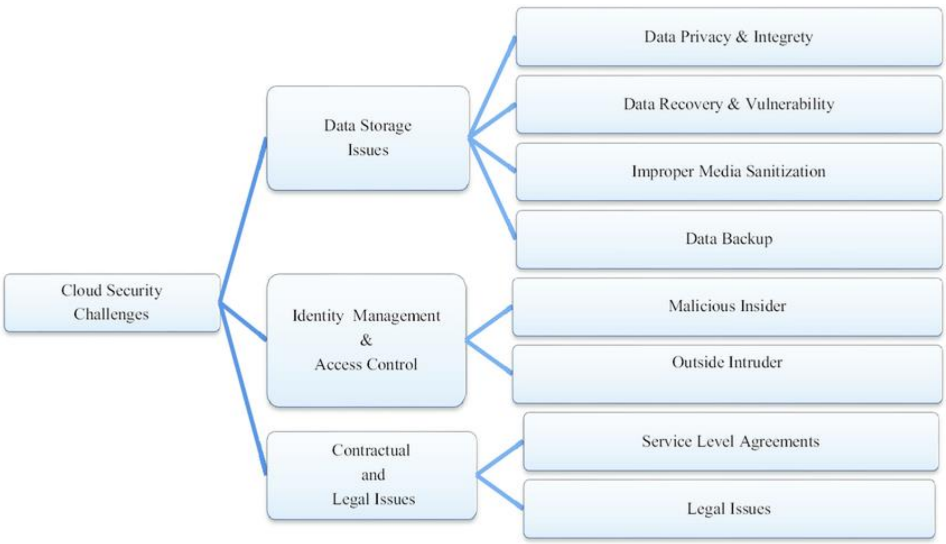


Figure 2.12: Cloud Security challenges

CHAPTER – 3

LITERATURE REVIEW

3.1 LITERATURE SUMMARY

The advancement in path planning technologies in order to provide an optimal path has put human efforts in navigation at rest. Due to rapid economical growth and changing working conditions, the time on people's hand has reduced immensely. Thus for a newbie in a city or person going outside the city by road, wasting time in prior path planning is futile. There has been many researches in the past and are still going on intensively in the field (area) of optimized energy efficient path planning for various purposes like vehicle routing, disaster management, routing of autonomous vehicles, robot movement in games and otherwise, UAV movement etc. All these researches show various algorithms and techniques for an efficient path planning.

This chapter further highlights those techniques, the comparison between them and their involvement in path planning and optimization. It also represents a summation of a few of the past researches in this field.

Aimi Najwa Sabri et.al.,2018 [1], the aims to compare the traditional algorithm namely A* Algorithm and a metaheuristic search algorithm namely the Bee Algorithm, especially when used in the gaming environment for path finding. The author has first described the both the algorithms in brief along with the small introduction of the Dijkstra Algorithm. The comparison of both the algorithm was done based on two factors i.e. time and memory required to reach the goal node. This was done on different size grid maps with different level of complexity. As the result it was found that the A* Algorithm searches the path 10 times faster than the Bee Algorithm when there is a free obstacle game environment. However Bee Algorithm is more efficient in searching routes though more complex game environment.

Mina G. Sadek et.al., 2018 [2], aim was to introduce a sensor-based approach for online coverage path planning problem in order to get an optimal solution. The terms online means that the coverage area is unknown. Here the authors have made use of two techniques that are: (i) Multi-objective optimization genetic algorithm (GA) and Dynamic Programming (DP). First, the proposed approach divides the main problem of finding a complete coverage path into many sub-problems with the aid of Dynamic Programming followed by individually solving these sub-problems by treating them as main problem using multi-objective genetic algorithm. Keeping path planning of Coverage region (PPCR) algorithm as a benchmark the proposed algorithm showed more efficiency in very complicated environment with more complex obstacles, while in less complex environment the performance of both the algorithms were close enough.

Hironori Hiraishi, (2017) [3], researched a way to enhance the performance of the autonomous vehicles namely self driving car by creating a routing method that is adaptable to the customers mood. The authors have stated that it not always best to take an avoidance route where traffic is very low. Sometimes driving along the current route is more efficient or a faster way than to create and avoidance route. Due to this issue the author proposed a method where the decision to take the avoidance route or the current one depends on the mood of the customer sitting in the car. If the customer is relaxed and comfortable the car will take the current route, if the customer is tired or agitated then the car will reach the destination earliest as possible by avoiding heavy traffic and if the customer is exhausted the vehicle will stop for a break period. In order to detect the mood of the customers the authors have utilized sitting pressure sensor which recognizes the customer fatigue which in turn could predict the mood of the customer. While the route finding method used is Time-constrained heuristic search which allows the vehicle to obtain an avoidance route without stopping prior to entering the traffic.

Emna Mejri et.al. (2017) [4], aims at finding an energy efficient solution to autonomous electric vehicles. This approach states that the energy loss is not only because of longer distance but due to the rolling resistance when the vehicle is at low speed. The obstacles on way are also one of the several reasons for higher energy consumption. Thus in to create an energy efficient path planning method Dijkstra Algorithm is used along with the environment map which includes the workspace, a map for environmental rolling resistance and the dynamics of the vehicle. As a result the research was able to give an energy - efficient path planning technique for the vehicles and also it proved that shorter paths to the destination are not always the answer to energy efficiency problems.

Hu Pan et.al (2017) [5], aim was to improvise on the traditional A* algorithm's efficiency in a known environment when it it comes to robot path planning. The researched reviewed various disadvantages of A* such as the redundant and inflection points in the path of the robot that it can easily deviate from. Because of this flaw a new advanced version of A* algorithm was introduced in this paper. This was done by adopting two methods. The first was to lessen the redundant points by a smoothening method followed by the reduction of inflection points by weighting the evaluated function in A*. Also there was an addition of the sub-node generation strategy with the objective of avoiding the obstacle. The authors were successful in improving the efficiency of traditional A* algorithm.

Zhibin Nie et.al (2016)[6], purpose was to overcome the shortcomings of the basic Particle Swarm Optimization (PSO). The stated shortcomings were weak local search ability and poor updating formula. In the proposed research the inertia weights were adjusted dynamically by combining simulated annealing and particle swarm algorithm. This removed the problem of PSO that it is easily trapped in the local optimum and hence the global optimal and near-optimal solutions cannot be obtained. The results shows that nonlinear inertia weight PSO is better than the traditional PSO algorithm in both simple and complex grid environments. The stimulated annealing PSO also shows very good results in complex environment.

E. E. Ogheneovo and E. Seetam (2016) [7], aim was to determine the shortest and the most cost-effective routes. This research used dijkstra algorithm to find the smallest routes while and heuristic search graphs are used to find a cost-effective solution. The proposed algorithm focused in developing a graph based algorithm for solving the problem of route optimization especially in Nigeria tourism. It provides a model focused on synthesis of heuristic graph and adaptive routing. When compared to the other routing algorithms the proposed method showed more better results in efficiency.

Yi-zhou Chen et.al (2014) [8], aim was to develop a dynamic road network model for vehicles evacuation during emergency events, like, earthquakes, hurricanes, fires, terror attacks, accidents etc. All these events may lead to loss of life and health of people. The proposed research used dijkstra algorithm that provides optimal evacuation path in three different cases namely morning peak, common and evening peak. The results produces an efficient theoretical basis and a well predictive method for optimal path selection for emergency evacuation and emergency rescue decision in public places mostly where population density is very high.

Wang Yu-qin and Yu Xiao-peng (2012) [9], has dome this research to improve the search ability of robot path in an unknown environment by avoiding obstacles and reach the destination in less time. The process to obtain the above aim was as followed: (i) the robot two- dimensional space model was created by MAKLINK which is a classical method in graph theory for modeling free spaces and finding optimal solutions in the entire planning space. (ii) After the creation of robot two - dimensional space model the problem was transformed into a shortest path problem which was solved using Dijkstra Algorithm. (iii) Finally, the path optimization was done through a new and improved version of Particle Swarm Optimization (PSO) called the Immune Particle Swarm Optimization (IPSO) which is based on immune system's biological mechanism. As a result the introduced IPSO along with the entire process has improved the overall efficiency, the convergence speed and the robustness of time-varying parameters.

Jinhao and Lu Chi Dong.(2012) [10], aim was to enhance the overall performance of the traditional Dijikstra Algorithm. This research focuses on the drawbacks of the dijkstra algorithm and has worked to optimize it. The optimization was done in three fields namely, selection of nodes of the shortest path, data storage structure and organization. This paper mainly studies the various application of shortest path algorithm based on data structure. As a result, the optimized dijkstra has improved space complexity, time complexity. It has also reduced the storage space, data redundancy and has drastically improved the running rate.

Bo Huang et.al. (2011) [11], aims at providing an improved route optimization by improving dijkstra algorithm. It is done by researching on the memory occupation and time consumption in searching of the algorithm. As the result of this research there was a great improvement in the search time as it saves half of the time for every search.

LI Guang-ru and ZHi Sun (2010) [12], provide an intelligent system that takes in real-time traffic information through GPS and solves the problem scheduling container trucks. The proposed work has used optimal capability of ant colony algorithm by putting information entropy into it, in order to improve the efficiency.

This is done for the searching process of food resources. This model is capable of providing real -time dynamic scheduling scheme to solve the problem of dynamic doc scheduling.

Low, C.Y. et.al (2010) [13], aim was to method capable of optimizing the path of gantry robots and providing more efficient and economic movement. It also aims at shortening the overall travel length. The algorithm used in this approach was Genetic Algorithm (GA), which is a robust optimization technique that belongs to the class of stochastic search methods[2]. In order to obtain the best and efficient solution for the path optimization of the gantry robot, the genetic parameters are adjusted by developing a well suited fitness function and GUI Simulation. The result of this research were very positive as the motor turn on and the robot travelling time was shortened.

Tianmiao Wang et.al. (2008) [14], proposed two approaches to solve the staying-alive and energy-efficient path planning problem. Here the authors want to create a method through which a robot is able to complete its task and return to the docking station for battery with the amount of energy that it has. This is the dynamic energy-evaluation technique where it is considered that the robot has enough energy to finish all its task and return to the charging station. The two approaches based on greedy Travelling salesman problem (TSP) and Tabu-search method were proposed for staying alive and energy-efficient path planning. The outcome of the research showed that Tabu-search Based approach was best and efficient. It was capable of providing an effective path planning method with which the robot was guaranteed to stay alive and finish all its task with minimum energy.

Geethu Elizebeth Mathew(2015) [15], proposes a new method that is capable of generating a higher quality path using less time and memory as compared to other existing solutions in video games. It also reviews all the existing widely used pathfinding solutions. The author has introduced a new heuristic method for increasing the speed of pathfinding process on a map represented using grids with uniform cost. The most important part of this approach is the identification of the nodes to be explored and proper usage of the data. As a result this direction based heuristic approach enhance the speed and the efficiency while reducing the memory consumption. This is because the number of node to be explored are limited and are carefully selected. The significance of this method is in the video games.

Jose A. Mocholi et.al. (2010) [16], aim is to determine a method capable of taking into account the emotions of the game characters in path finding. This is done because the games these days are more visually realistic graphics which increases the realism of the characters in the game. The Ant colony Optimization (ACO) algorithm is used for decision making that also takes into account the emotions needed to give a more optimal decision. The results concluded that ACO was capable of improving the pathfinding process in games having emotional behavior as the key element.

Zhou yefu Li and zaiyue (2010) [17], presents an approach to construct vehicle routing problem algorithm and analysis of its operation and results. The Algorithm used in this approach is Genetic Algorithm. The paper states that the problem of logistics delivery route optimization is an NP hard problem. Thus this paper helps in designing the optimized delivery routes with the help of Genetic Algorithm encoding which applies the powerful numerical computing function of MATLAB. The result of this research showed that genetic algorithm proved to a good performance heuristic algorithm in searching optimal logistics delivery routes. This can provide an efficient solution to logistics delivery route quickly and effectively.

In the past researches, there are a variety of algorithms used along with their multiple variant for different applications. Table 1, shows the summary of some of these algorithms.

TABLE 3.1

THE SUMMARY OF ALGORITHMS USED IN SOME PAST RESEARCHES

S.NO.	AIM	ALGORITHM USED	FINDINGS	REFERENCES
1.	The basic aim of this paper is to compare the traditional Pathfinding algorithm namely, A* Algorithm with the search optimization algorithm namely, Bee Algorithm	Bee algorithm and A* Algorithm	A* algorithm is 10 times faster than the Bee Algorithm when there are no obstacles in the game but when the game is complex Bee is better option.	Aimi Najwa Sabri et.al.,2018 [1]
2.	Aim of this paper is introduce a sensor based approach for coverage path planning to produce optimal results using the combination of Dynamic Programming and Genetic Algorithm	Dynamic Programming (DP) and Genetic Algorithm (GA)	Keeping the PPCR (Path planning of coverage region) as a benchmark the proposed approach was more efficient in complex environments whereas in the environment with less obstacle the results of the both the approaches was similar	Mina G. Sadek et.al., 2018 [2]
3.	The aim was to develop a route planning method for self-driving cars based on the customers mood	Time - constrained Heuristics search (TCS)	It was found the TCS was more efficient in finding traffic avoidance route than A* even before entering the congested area and	Hironori Hiraishi, (2017) [3]

			to make the algorithm more efficient the avoidance route was taken based on the customer's mood.	
4.	To develop an energy efficient path planning technique for the autonomous electric vehicles.	Dijkstra Algorithm	The research was able to prove that the shorter paths are not always the energy-efficient solutions but the area and vehicle dynamics also plays an important role. It was also able to give an energy efficient path planning technique.	Emna Mejri et.al. (2017) [4]
5.	To improve the efficiency of traditional A* Algorithm	A* Algorithm	The overall efficiency of the A* was improved in the research by removing the two disadvantage of A* namely redundant and inflection points.	Hu Pan et.al (2017) [5]
6.	The aim was to overcome the drawbacks of Particle Swarm Optimization (PSO)	Non-linear inertia weight PSO and Stimulated annealing.	The combination of the non-linear inertia weight PSO and stimulated annealing was more efficient in autonomous robot path planning than traditional Particle Swarm Optimization (PSO)	Zhibin Nie et.al (2016)[6]
7.	To find the	Dijkstra	It provides a model	E. E. Ogheneovo

	shortest and cost effective routes in Nigeria tourism	Algorithm and heuristic graph search technique	focused on synthesis of heuristic graph and adaptive routing. When compared to the other routing algorithms the proposed method showed more better results in efficiency.	and E. Seetam (2016) [7]
8.	To provide a dynamic road network model for determining an optimal vehicle evacuation path during a disaster or emergency.	Dijkstra Algorithm	The results produces an efficient theoretical basis and a well predictive method for optimal path selection for emergency evacuation and emergency rescue decision in public places mostly where population density is very high.	Yi-zhou Chen et.al (2014) [8]
9.	To improve the efficiency of the robot based path planning technique	MAKLINK, Dijkstra Algorithm and Immune Particle Swarm Optimization (IPSO) Algorithm.	The IPSO proved to be very efficient and feasible in path planning. It, along with the other processes improved the convergence speed robustness of time varying parameters.	Wang Yu-qin and Yu Xiao-peng (2012)[9]
10.	To optimize the traditional Dijkstra algorithm for calculating shortest path	Dijkstra Algorithm	The results clearly showed a drastic improvement in the performance of dijkstra algorithm in terms of improved space	Jinhao and Lu Chi Dong.(2012) [10]

			complexity, time complexity and running rate followed by reduction in storage space and data redundancy making it more applicable to calculate the shortest path.	
11.	Aims at providing an improved route optimization by improving dijkstra algorithm.	Dijkstra Algorithm	The results showed great improvement in the search time as it saves half of the time for every search.	Bo Huang et.al. (2011) [11]
12.	Aims at providing an intelligent system that takes in real-time traffic information through GPS and solves the problem scheduling container trucks	Ant Colony Algorithm	This model is capable of providing real - time dynamic scheduling scheme to solve the problem of dynamic doc scheduling.	LI Guang-ru and ZHi Sun (2010)[12]
13.	The aim was to method capable of optimizing the path of gantry robots and providing more efficient and economic movement.	Genetic Algorithm	The result of this research was very positive as the motor turn on and the robot travelling time were shortened.	Low, C.Y. et.al (2010) [13]
14.	The authors proposed two approaches to solve the	Greedy Travelling Salesman Problem (TSP)	The outcome of the research showed that Tabu-search Based approach	Tianmiao Wang et.al. (2008) [14]

	staying-alive and energy-efficient path planning problem	and Tabu - search	was best and efficient.	
15.	A method that is capable of generating a higher quality path using less time and memory as compared to other existing solutions in video games.	Direction based heuristic search.	Direction based heuristic approach enhance the speed and the efficiency while reducing the memory consumption.	Geethu Elizebeth Mathew(2015) [15]
16.	The aim is to find a method capable of taking into account the emotions of the game characters in path finding.	Ant Colony Optimization (ACO)	The results showed that ACO was capable of improving the pathfinding process in games having emotional behavior as the key element.	Jose A. Mocholi et.al. (2010) [16]
17.	Presents an approach to construct vehicle routing problem algorithm and analysis of its operation and results.	Genetic Algorithm	The result of this research showed that genetic algorithm proved to a good performance heuristic algorithm in searching optimal logistics delivery routes.	Zhou yefu Li and zaiyue (2010) [17]

3.2 CONCLUSION

According to the survey done on various past researches, there are various methods, techniques and algorithms that can be employed in path planning and optimization for various purposes. For purpose of vehicle routing the energy-efficient optimized path can be very conveniently found with the help of Dijkstra algorithm [4]. Even in the case of vehicle evacuation during emergency situation Dijkstra can give the fastest results [8]. But such is not always the case. In situations where there is a complex environment with many obstacles dijkstra does not give an efficient result. The combination of PSO and stimulated annealing gives and efficient way for finding shorter and smoother paths in both complex and simple environment than the traditional PSO[6] Cost effectiveness of a particular path can be calculated using heuristic graph search while the shorter paths can be calculated using Dijkstra algorithm which could be very useful in effective path planning and optimization [7]. In environment having many obstacle and complex setting an optimized path finding can be done very efficiently using Bee Algorithm especially for game navigation [1].For areas where full coverage is required and the region is unknown such as a newly developed area or new city the path planning can be done with the combination of Dynamic Programming and Genetic Algorithms to produce better results [2].Time - constrained heuristic search is more efficient in traffic avoidance than the A* .The decision to whether avoid the traffic or not based on the customer mood and fatigue could enhance the efficiency of the self driving cars [3]. There also various other algorithms and their application in the context of path planning. Some can be used as it is while other can be enhanced to fit the scenario or problem.

CHAPTER – 4

PROPOSED WORK

4.1 PROPOSED WORK

The use of applications like Google maps gives an individual a shortest path with lesser time and traffic. However such systems provide only a single route from the source to destination without giving any further choices that could help a user plan out a path according to its resources and necessities. But such is not always the situation.

Some time the path planning is more driven towards the resources available on that path or the fuel in the vehicle. A real time example of such a situation is that of a pantry truck. Such trucks carry fruits and vegetable that could get spoiled after a certain time or if the preservation conditions are not met. In this case the requirement is not only of less time, distance and traffic but also of a route that has pantry warehouses in between source to destination, so that if there is a fault in the truck or any other calamity these fruits and vegetables can be preserved. Similarly, if a person wants to go home from work but also needs to buy groceries than an approach that will help the user to decide a path with more grocery stores is more efficient in path planning.

This proposed work, improves the ability of already existing researches on path optimization. It aims at developing a more efficient path optimization algorithm that helps in overcoming the major drawbacks of the previous systems and could prove to be more helpful in many similar above mentioned real time examples. The objective is not only to focus on the three factors namely time, distance and traffic but to include energy consumption and availability of resources too. It also focuses on finding out multiple routes from the source to destination, giving a user various path options to decide from, according to the current need. This all will be done through well trained algorithm and data sets generated based on past experiences and user's feedback.

4.2 AIM OF THE RESEARCH

The work that have been done in the past for finding an optimized path were based on time, distance and traffic. Furthermore the result of such an optimization is a single route from the source to destination without any choices. A user is not provided with options to plan out a path according to the need of resources and available fuel. This problem is what leads to an attempt to do research of an approach that provides multiple optimized path options based on five factors i.e. time, distance, traffic, fuel consumption and resources available for a particular source destination pair.

This research aims on achieving numerous advantages such as the ability to navigate an area without physically planning the route on the map of that area or city or manually travelling on roads which will not only eliminate extra human effort but also save a lot of time. It also aims at saving fuel, by estimating the total fuel consumption of all possible routes for a particular source destination pair. The key advantage of this approach is its potential to navigate routes according to the need of specific resources, simultaneously providing multiple route options to choose from.

4.3 OBJECTIVE

1. Analyzing various routes in order to provide multiple path options for a particular source destination pair to provide more efficient path optimization algorithm, based on five major factors:
    - ❖ Time, that will help reach a destination in lesser time ,
    - ❖ Distance, that provides a shortest route between two points,
    - ❖ Traffic, that hints on the amount of vehicles travelling those paths at that instance of time
    - ❖ Energy, that provide an insight on fuel consumption of each path options and lastly

- ❖ Resources, that helps an individual to plan a path according to its current needs like the path having more medical stores if person needs to pick medicines on the way to the destination.

2. Presenting the above findings on a map for better understanding.

4.4 ALGORITHMS USED

4.4.1 FLOYD–WARSHALL ALGORITHM

The Floyd–Warshall algorithm, in computer science, is used mostly for finding shortest distances in a weighted graph with both positive or negative edge weights (but with no negative cycles).

In a single execution of the algorithm, the lengths (summed weights) of shortest paths between all pairs of vertices can be found. Even though it does not return details of the paths themselves, it is possible to reconstruct the paths with simple modifications to the algorithm.

Figure 4.1, shows an example of Floyd – Warshall Algorithm where the list of shortest paths is calculated from source to destination.

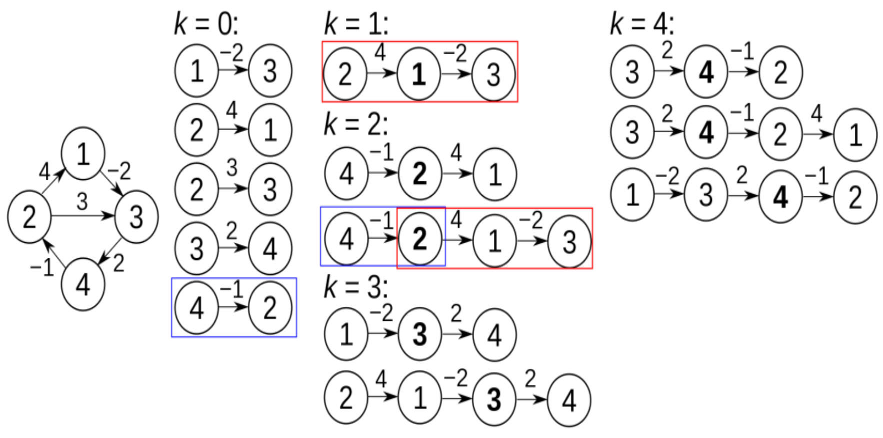


Figure 4.1: An example of Floyd–Warshall algorithm

4.4.1.1 HOW DOES FLOYD – WARSHALL ALGORITHM WORKS?

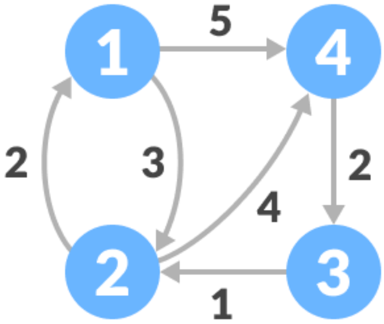


Figure 4.2: Initial graph

Figure 4.2, shows the initial given graph.

Following steps are followed to find the shortest path between all pairs of vertices.

1. A matix A^0[][] of dimension n x n is taken, where n represents the number of vertices.
2. In each cell of the given matrix A^0[i][j], the distance from vertex i^{th} to the vertex j^{th} are filled. If no direct path exist between the i^{th} and the j^{th} vertex, then that cell is filled as infinity (Figure 4.3).Also all the diagonal elements in the matrix are filled as 0. This is represented as the A^0, or the initial matrix.

$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 5 \\ 2 & 0 & \infty & 4 \\ \infty & 1 & 0 & \infty \\ \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

Figure 4.3: Each cell is filled with the distance between i^{th} and the j^{th} vertex.

3. A matrix A^k is created , where k is the first vertex i.e. 1. Since value of k is 1, thus the first row and first column elements are left as it is and the diagonal elements are filled as 0.

The rest of the elements of matrix $A^1[i][j]$ is filled using A^0 matrix as follows:

- ❖ (A[i][k] + A[k][j]) if (A[i][j] > A[i][k] + A[k][j]), which means if the direct distance between source and destination is bigger than the path following vertex k, then the value of cell becomes (A[i][k] + A[k][j]) (Figure 4.4)

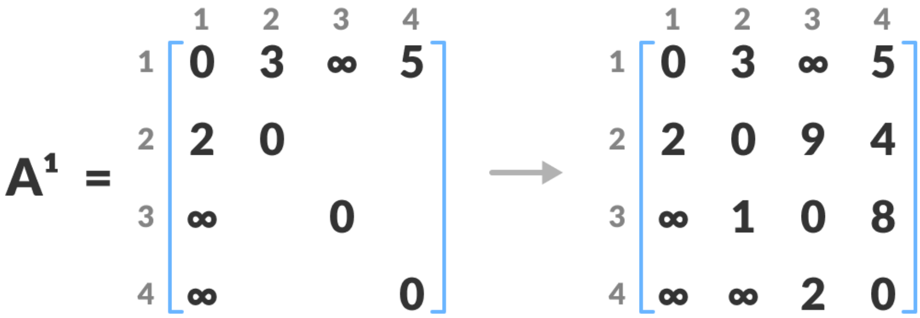


Figure 4.4: The values of all the elements of A^1 matrix are filled using A^0 matrix

The above step is repeated for all the other vertices in the graph.

4. For vertex k = 2, matrix A^2 is created. Since value of k is 2, thus the second row and second column elements are left as it is and the diagonal elements are filled as 0.

Then using matrix A^1, step 3 is repeated to create matrix A^2. (Figure 4.5)

$$A^2 = \begin{bmatrix} 0 & 3 & & \\ 2 & 0 & 9 & 4 \\ & 1 & 0 & \\ & \infty & & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 3 & 9 & 5 \\ 2 & 0 & 9 & 4 \\ 3 & 1 & 0 & 5 \\ \infty & \infty & 2 & 0 \end{bmatrix}$$

Figure 4.5: The values of all the elements of A^2 matrix are filled using A^1 matrix

5. For vertex k = 3, matrix A^3 is created. Since value of k is 3, thus the third row and third column elements are left as it is and the diagonal elements are filled as 0.

   Then using matrix A^2, step 3 is repeated to create matrix A^3. (Figure 4.6)

$$A^3 = \begin{bmatrix} 0 & & \infty & \\ & 0 & 9 & \\ \infty & 1 & 0 & 8 \\ & & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 3 & 9 & 5 \\ 2 & 0 & 9 & 4 \\ 3 & 1 & 0 & 5 \\ 5 & 3 & 2 & 0 \end{bmatrix}$$

Figure 4.6: The values of all the elements of A^3 matrix are filled using A^2 matrix

6. For vertex k = 4, matrix A^4 is created. Since value of k is 4, thus the fourth row and fourth column elements are left as it is and the diagonal elements are filled as 0.

   Then using matrix A^3, step 3 is repeated to create matrix A^4. (Figure 4.7)

$$A^4 = \begin{bmatrix} 0 & & & 5 \\ & 0 & & 4 \\ & & 0 & 5 \\ 5 & 3 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 3 & 7 & 5 \\ 2 & 0 & 6 & 4 \\ 3 & 1 & 0 & 5 \\ 5 & 3 & 2 & 0 \end{bmatrix}$$

Figure 4.7: The values of all the elements of A^4 matrix are filled using A^3 matrix

7. Thus the matrix formed using the last vertex (i.e. k = 4) in the graph, in this case it is A^4, for each pair of vertices, gives the shortest path.

4.4.2 0/1 KNAPSACK ALGORITHM

The Knapsack problem also called the rucksack problem. It is a problem in combinatorial optimization, which means it aims at finding an optimal solution for a given limited set of problems. It states that when a set of items are given, each of which have a weight and a value, the number of each item to involve in a collection is to be determined so that the entire weight is less than or similar to a provided limit and the total value is as big as possible.

It's name is derived from a problem faced by someone who has to choose and fill the most valuable items but is constrained by a fixed-size knapsack .

In 0-1 Knapsack, there is no possibility of breaking the items which means that either the items should be taken as a whole or should be left. This is reason behind calling it as 0-1 Knapsack, which means either 1(whole) or 0 (nothing).

Hence, in case of 0-1 Knapsack, the value of x_i which is the item can be either ***0*** or ***1***, where other constraints remain the same.

Figure 6, shows an example of a one-dimensional 0/1 knapsack problem: which boxes should be chosen to maximize the quantity of cash while still keeping the overall eight under or up to 15kg? This problem could take both the eight and volume of the boxes. (Solution: if any number of boxes can be chosen, then three yellow boxes and three grey boxes; if only the presented boxes are allowed to choose from, then all the green boxes.)

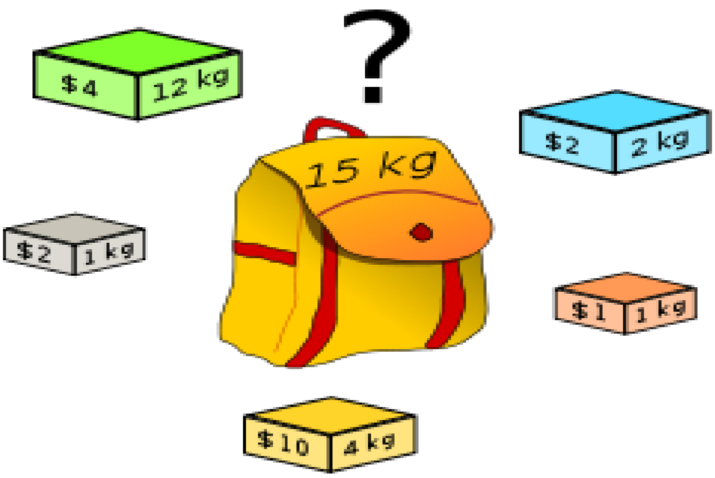


Figure 4.8: Example of 0/1 knapsack problem

4.5 METHODOLOGY

1. Analysis of all the different paths for multiple locations along with the resources available on those paths in order to create data sets.
2. Creation of a trained data set of users feedback. This feedback will be taken based on all five parameters (i.e. Time, Distance, Traffic, Resources and Fuel/energy) that are being used to calculate all the optimal paths.

3. Development of an algorithm which will process the requirement of the user using LSTM Algorithm under the dijkstra and 0/1 Knapsack algorithm to achieve an optimal result.
4. Presentation of the generated results on the map in a route/path format.

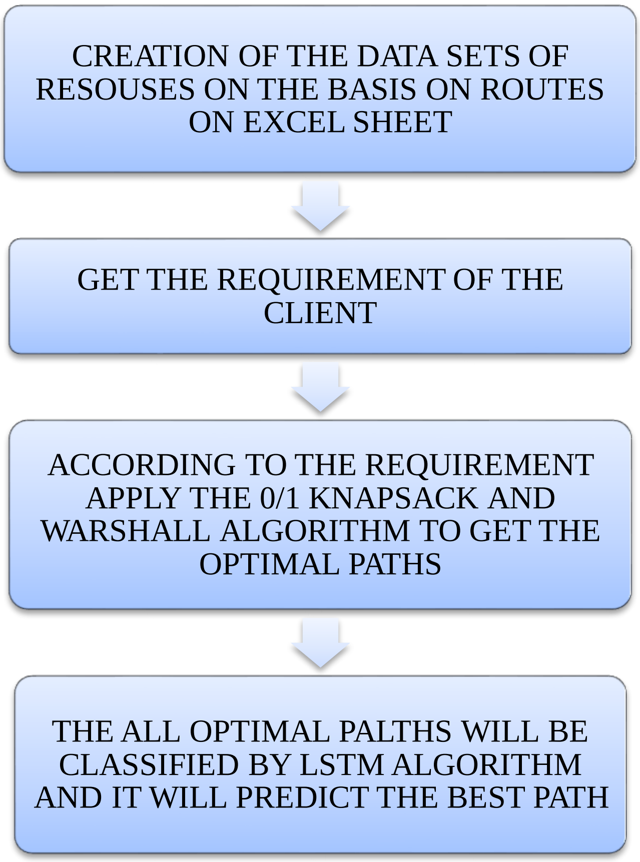


Figure 4.9: Flow chart of methedology

4.5.1 FLOWCHART

Figure 4.10, shows the flowchart of the process followed to implement the work module. According to this flowchart:

At the beginning, the location of the user will be recorded through GPS which will become the source, along with the requirements (such as resources, destination etc.)

After the selection of the starting point and the destination, the multiple path between them will the recorded in the form of matrix (along with all the nodes, i.e. the resources location on the paths and their values i.e. the distances) using the floyd - warshall algorithm .

Then taking the all the nodes, which is the resource requirement entered by the user, available on the most optimal path, the time travel distance xa between each two nodes is calculated. For all node path, initially, xa = 0, and if there is no direct connection between the two nodes then, for rest all the nodes, xg = ∞. The values for each node are also checked.

Continuing on the path, if there are no left or right turn around a particular node, then the straight path is followed upto the next node and three components are recoded V-L-W (i.e. value, time and distance). The next node becomes the virtual node L' and is added to x. If there is a left or the right turn then the nodes available on that paths are also check for V-L-W.

The two matrix are created based on the above collected data, D[][] and V[][], where D[][] is based on minimum distance among nodes and V[][] is based on maximum values among nodes.

An $x^{(m)}$,where m represents the row and column, is computed for each matrix (1) and (2) using triple operation, at every $x^{(m-1)}$ point. This process is done following the 0/1 knapsack algorithm so that maximum resources could be covered using the shortest travel time.

Then finally all the x(m) are compared to find one optimal path capable of covering maximum resource requirement provided by the user in the minimum possible time and distance.

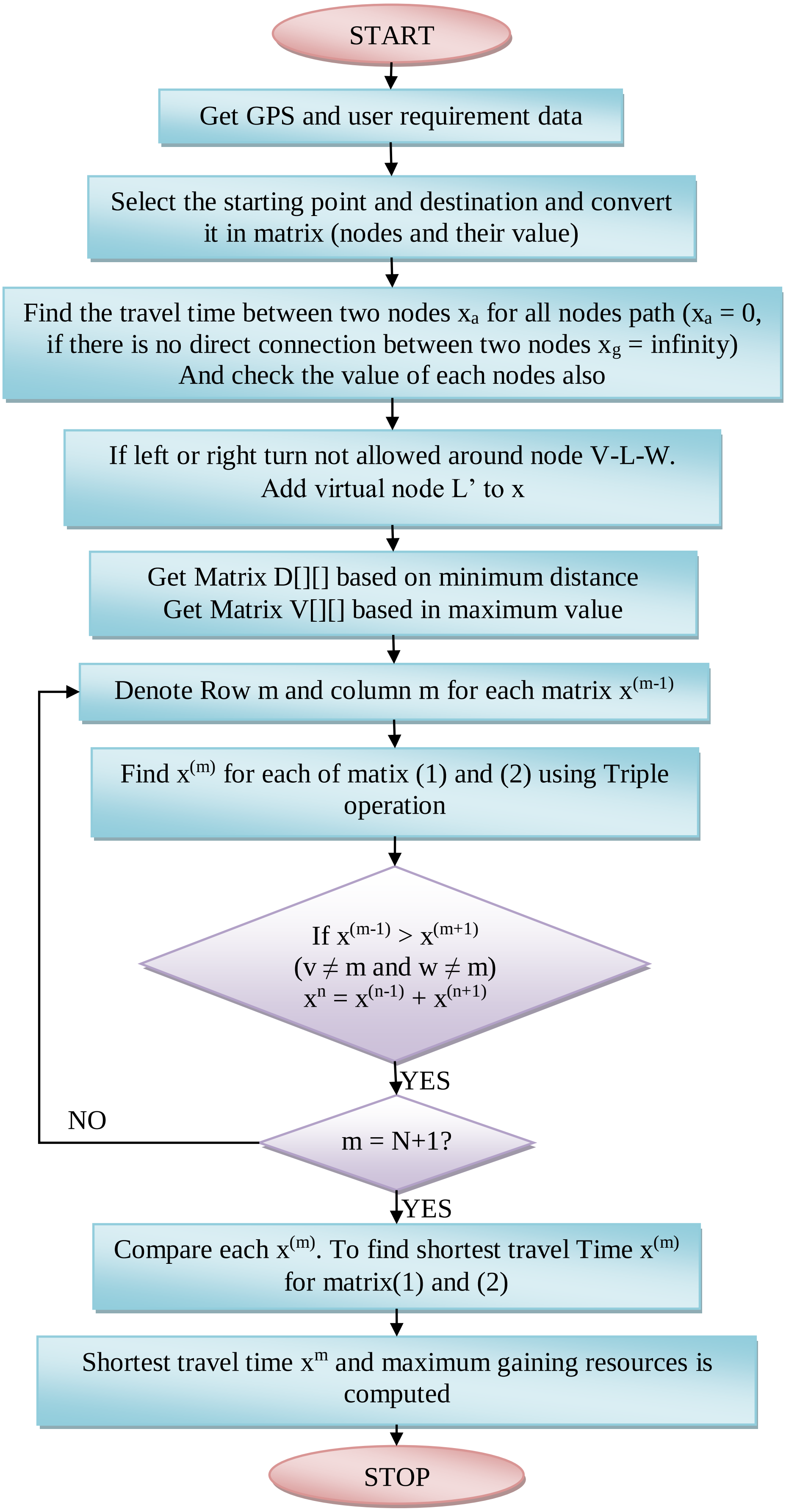


Figure 4.10: Flowchart explaining work module

4.5.2 ALGORITHM USED:

```
1)  let distance be D[n] x [n] and  value be V[n] x [n]
2)  for matrix of distance initialized to minimum distance initialized or ∞ and
    value to maximum value gaining.
3)  for each edge (u, v) do
4)      D [u][v] ← distance
5)      for each vector v do
6)          V [u][v] ← Value
7)      end for
8)  end for
9)  for m from 1 to v
10)     for x from 1 to v
11)         for y from 1 to v
12)             if D[x][y] > D[x][m] + D[m][y]
13)                 D[x][y] ← D[x][m] + D[m][y]
14)                 Value ← V[x][m] + V[m][y]
15)                 Pr ← Knapsack(V, x, y)
16)             end if
17)         end for
18)     end for
19) end for

    Knapsack (V, x, y)
20)     for w from 0 to x do
21)         c[0][w] ← 0
22)     end for
23)     for i from 1 to n do
24)         c[i][0] ← 0
25)     end for
26)     for i from 1 to w do
27)         if w_i ≤ value then do
28)             V ← value
29)         else
30)             V ← value[i+1]
31)         end if
32) return V
```

4.5.3 DESCRIPTION OF THE ALGORITHM:

In the proposed algorithm the aim is find an optimal path based on the resources and choices entered by the user. The user will be able to see all the paths available from the entered source to the destination, also will be given the most optimal path which use less distance, time fuel etc. The algorithms and its explanation is given further in this chapter

When the user enters the information of the source and destination it wants to travel and the resource which it need along the travelled path, all the valid paths from the source to destination will be shown to the user. User has a choice to either choose the path of its interest or will also be given a suggestion for the best optimal path. This is done especially for the user who does new to the city and does not know the roads well.

Based on the flow chart in the previous chapter, for finding that optimal path the above mentioned algorithm comes into play. According to the algorithm,

- ❖ Firstly, two matrices are created for the distance and the other for the value or priority of the resources (step 1). Those matrices are represented as, D[n][n] for storing shortest distances for the resource mentioned by the user, on all the path options available from the source to destination.
- ❖ Then, initially in the D[][] matrix the minimum distance nodes are stored while the rest of the nodes are initialized to ∞. And the matrix of value V[][] is initialized to maximum value gained for each node (step 2).
- ❖ For all the nodes, edge(u, v), which are the selected resource available on the paths from source to destination, there distance and values are stored in the distance matrix, D[u][v], and the value matrix V[u][v] respectively, as shown from step 4 to 8.

- ❖ After this, according to step 9 to 15, for all the nodes in the distance matrix D[][], it is checked if the distance of the current node is greater than the next available node after the current node, then the distance of the next node is stored, the value of that node is stored in the value matrix V[][] and the function Knapsack() is called to implement 0/1 knapsack (step 20 to 32) i.e. rearrange the value matrix from highest to lowest, otherwise the distance and value for that node remains the same.
- ❖ The above step is done for all the nodes in the distance matrix D[][] and value matrix V[][].
- ❖ This is how the matrix two created and rearranged for both the distance and value. Based on the above matrix, keeping the selected resources as the priority followed shortest possible distance, minimum time and fuel, an optimal path is given to the user as a choice.

CHAPTER – 5

RESULT ANALYSIS

AND

DISCUSSION

5.1 RESULT ANALYSIS:

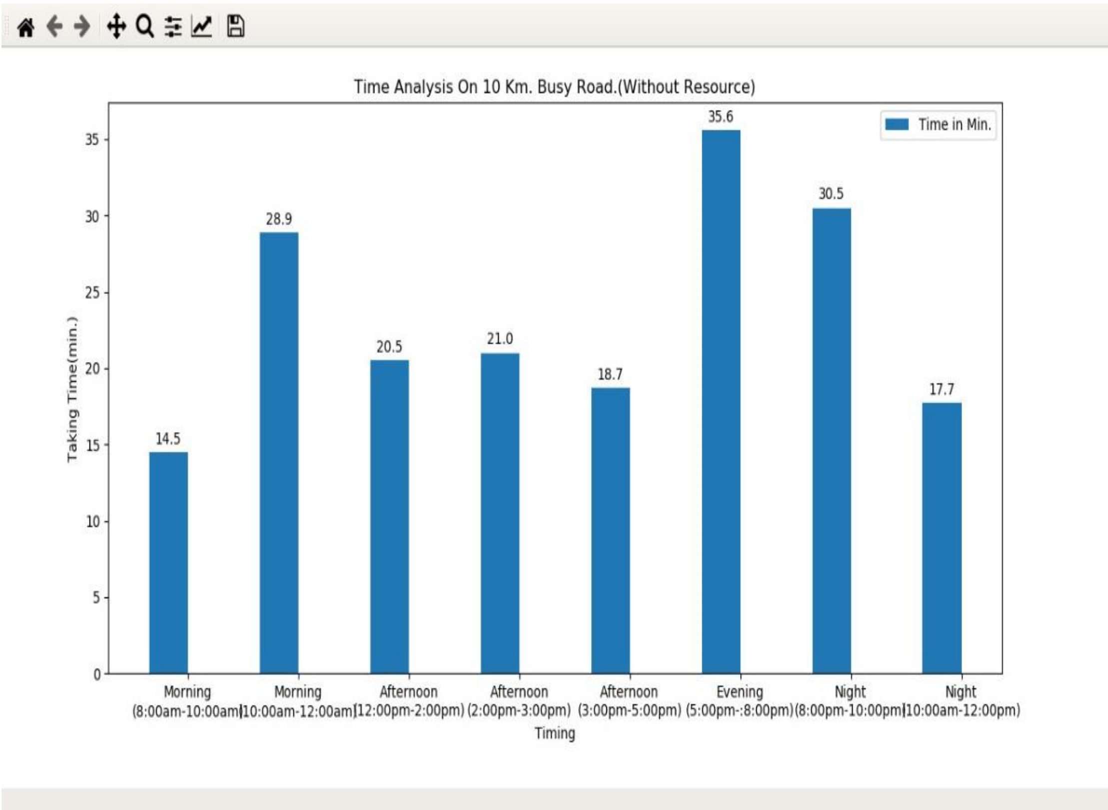
The algorithm mentioned above was implemented in various scenarios. The results and conclusions were derived:

A 10 km busy road was taken as an input for the navigation. This was done when there were no specific resources or locations needed to be accessed along the path. The road was navigated at different time period between 8:00 am (morning) to 12:00 am (night) and the time taken to travel that busy road at different time period was recorded. The details are shown in Graph 5.1, according to which, the minimum time taken to navigate the road is in the morning between 8:00 am to 10:00 am. During this time period i.e. morning, the traffic is very less even on the busy road. Thus it took only 14.5 minutes to reach the destination. Whereas the most time taken to travel the same road was during the time period between 5:00 pm to 8:00 pm, i.e. it took 35.6 minutes.

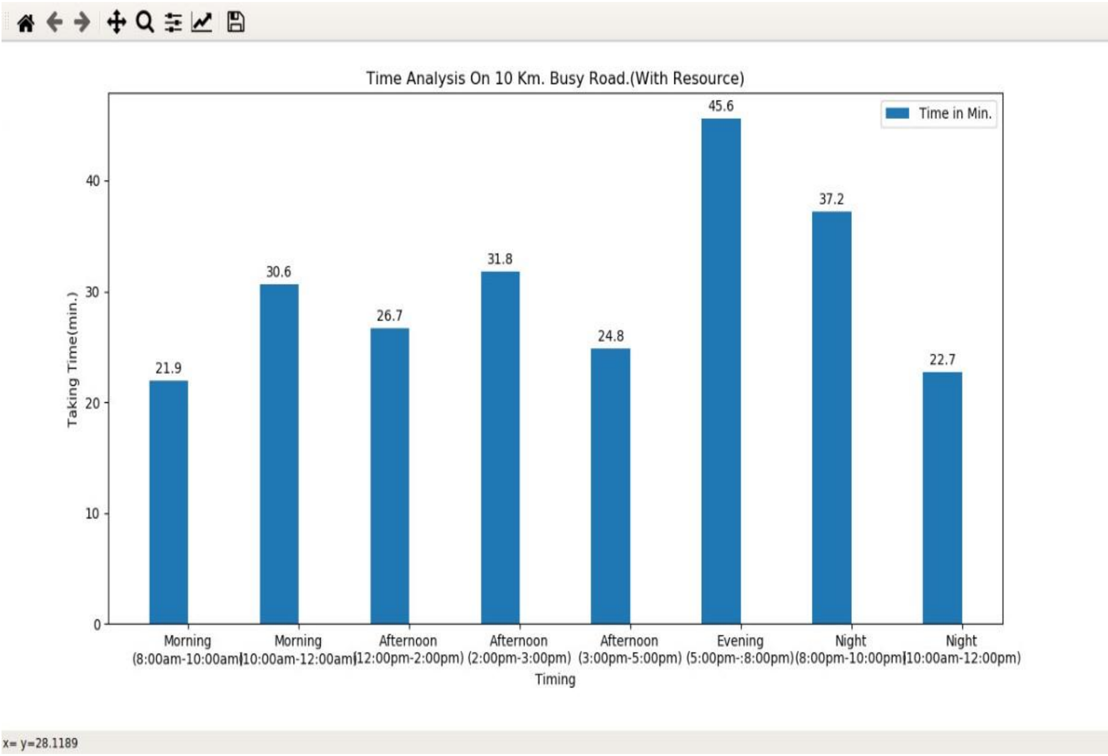
The same 10 km busy road was taken as input for navigation and this time the resource / location was chosen to visit along the path. Graph 5.2, shows the time recoded for the same time periods as Graph 5.1, when the road was travelled with resources. According to the graph, it shows that the minimum time taken even when the resource is chosen is during 8:00 am to 10:00 am in the morning, which is 21.9 minutes, while the maximum time taken to travel the same road with resources is 45.6 minutes, during 5:00 pm to 8:00 pm in the evening.

The comparative data of both the graph i.e. the time analysis of 10 km busy road with and without a resource choice is shown in table 5.1. Thus from the analysis, it is clear that the time taken to travel is more when trying to cover a chosen resource while it is less when travelled without any resource choice. But in either of the cases, whether a path travelled with the chosen resource on the way, or without any resource the maximum and the minimum time recorded to travel that road is 5:00 pm to 8:00 pm in the evening and 8:00 am to 10:00 am in the morning respectively.

This is due to the fact that in the morning there are very few people travelling the road, those of which are either going for jobs or for education, while in the evening there are people who are going for outing or functions long with those who are returning from offices. Hence the traffic predicted is more during the evening hours than morning. It also indicates that the ideal time for travelling is during morning hours. Thus all the above analysis further indicates that the algorithm chosen is efficient enough to provide satisfactory results in any case.



Graph 5.1: Time analysis on 10 km busy road, without resources



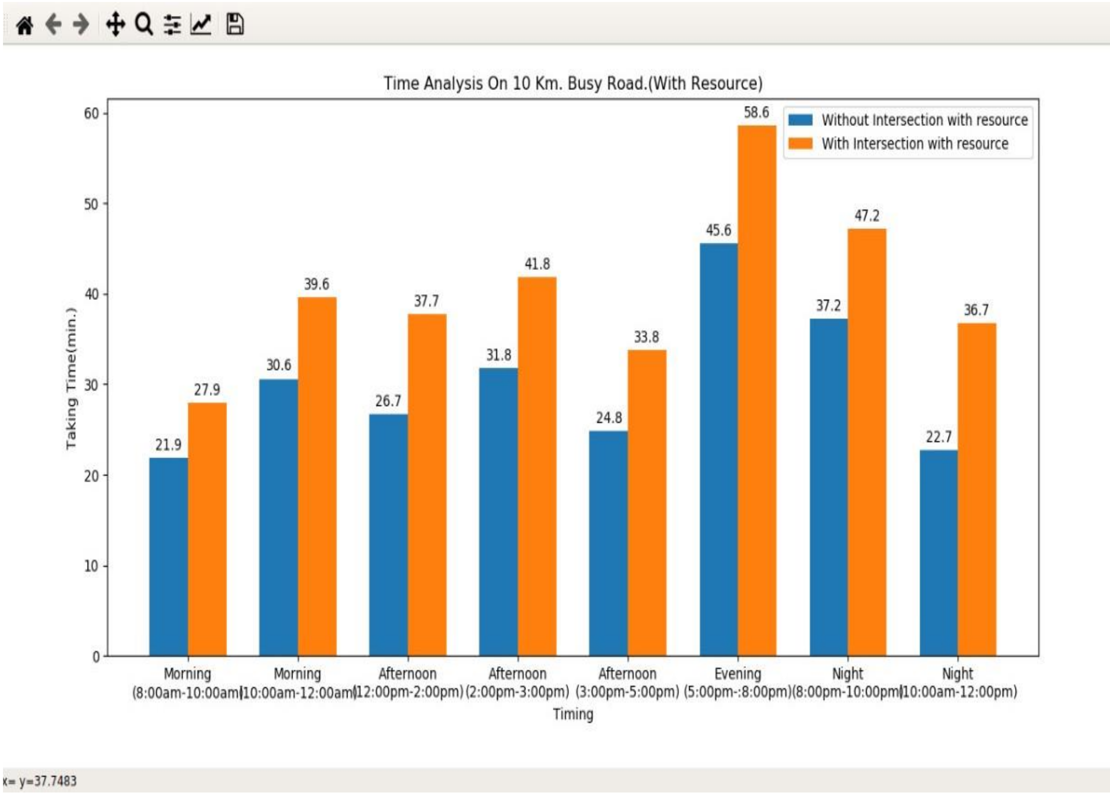
Graph 5.2: Time analysis on 10 km busy road, with resources

TIME	WITH RESOURCES	WITHOUT RESOURCES
8:00 AM – 10:00 AM	21.9	14.5
10:00 AM – 12:00 PM	30.6	28.9
12:00 PM – 2:00 PM	26.7	20.5
2:00 PM – 3:00 PM	31.8	21.0
3:00 PM – 5:00 PM	24.8	18.7
5:00 PM – 8:00 PM	45.6	35.6
8:00 PM – 10:00 PM	37.2	30.5
10:00 PM – 12:00 AM	22.7	17.7

Table 5.1: Comparison between the time recorded when travelling a 10 km busy road with and without resource

When following the algorithm it is shown that the optimal path cannot be the shortest path when the resources are also needed to be accessed along the path. Hence, only reducing the distance, time and fuel consumption is not enough but we also need a path which is capable of fulfilling the users need. Thus the ability to cover all the resources along the path is the main priority and to implement that there is the necessity of intersections. When during navigation if minimum distance and time is the priority then a shortest path from the source to destination is optimal, but in case of considering resources / locations/ stoppage along the path as per the traveler choice following the shortest path is not feasible as all the traveler’s requirements are not fulfilled by that path, also taking a longest path to fulfill those requirements defeat the purpose of path optimization. Hence, the middle solution is that to take intersections into consideration. When all the resources are not available along the path these intersections can be used to take turns in order to reach the resources and then get back to the same path using the next intersection.

Graph 5.3, shows the time recorded at different time period when travelling a 10 km busy road with the resource. It shows the comparison between two situations when the road is travelled following intersections and when taking a straight path. Table 5.2, shows the data derived from the graph. According to the comparison, it shows that the path followed using intersection take more time that the straight path. Even though it takes longer time to travel, it becomes important to take the path with intersections in order to cover all the resources if they are not available on a straight path.



Graph 5.3: Time analysis on 10 km busy road, with resources (with and without intersections)

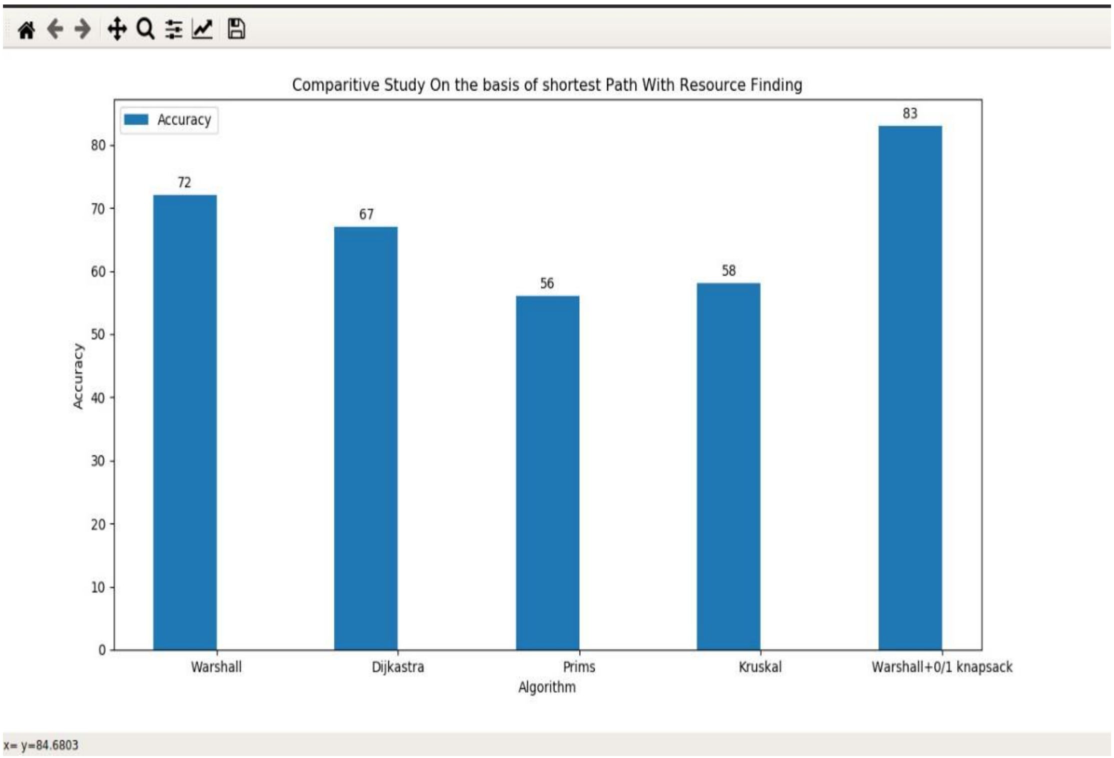
TIME	WITH RESOURCES	
	WITHOUT INTERSECTION	WITH INTERSECTION
8:00 AM – 10:00 AM	21.9	27.9
10:00 AM – 12:00 PM	30.6	39.6
12:00 PM – 2:00 PM	26.7	37.7
2:00 PM – 3:00 PM	31.8	41.8
3:00 PM – 5:00 PM	24.8	33.8
5:00 PM – 8:00 PM	45.6	58.6
8:00 PM – 10:00 PM	37.2	47.2
10:00 PM – 12:00 AM	22.7	36.7

Table 5.2: Comparison between the time recorded when travelling a 10 km busy road with resources, with and without intersections.

Lastly the above task of navigation was implemented using various algorithms, with purpose finding the most fasible algorithm to provide optimal result. The accuracy of all the algorithms were recorded as shown in graph 5.4 and table 5.3.

According to the graph and the table, when implementing Floyd-Warshall algorithm it showed 72% accuracy in providing multiple path from the source to destination. Rest of the algorithms that are dijkstra, prism and krushal were 67%, 56% and 58% accurate. Among which prisms has the lowest accuracy. Even though Floyd – Warshall was 72% accurate and is capable of giving multiple paths between source and destination but is still not enough to provide optimal results, where the traveler need other resources/locations to access along the path.

Hence, an algorithm was proposed which combines the features of Floyd – Warshall, which is capable of providing multiple paths from source to destination and 0/1 knapsack, which is capable of choosing resources based on priority and maximum value. The algorithm showed 83% accuracy when implemented to find a all possible paths from source to destination along with the choice of most optimal path which covers all the resource along with minimum possible distance, time and fuel.



Graph 5.4: Comparative study on the basis of shortest path with resource finding

ALGORITHM	ACCURACY
Floyd - Warshall	72%
Dijkstra	67%
Prism	56%
Krushkal	58%
Floyd – Warshall + 0/1 Knapsack	83%

Table 5.3: The accuracies of all the algorithm capable of finding shortest path with resource

5.2 DISCUSSION

According to Emma Mejriet.al, the most energy efficient solution for navigation of autonomous vehicles is done using Dijkstra algorithm [4]. This algorithm finds its use in optimization of various other computing devices. It has been used by Yi-zhou Chen et.al for determining vehicle evacuation path during disasters and emergency.The variants of Dijkstra have also been developed by combining it with heuristic search to find shortest and cost effective routes [7]. Whereas according to Wang yu-qin and Yu Xiao-peng, combining Dijkstra with immune particle swarm optimization helped to improve the efficiency of robot based path planning [9]. In the same way there are various other researches where Dijkstra has been proven to the most effective and commonly used algorithm for planning of shortest path. There are also various other algorithms which have come into use after dijkstra, such as A* , particle swarm optimization, genetic algorithm, ant colony algorithm etc. Now it has become a practice of combining classical path planning algorithms with the heuristic search algorithm.

Even though dijkstra has proved its efficiency in optimization, but where finding of multiple paths between two points is needed, it fails. Thus in this dissertation Floyd – Warshall algorithm was employed, which is capable of providing multiple paths between nodes. Using this ability of Floyd – Warshall it was combined with the 0/1 knapsack algorithm to further improve the efficiency of path navigation. Through this research it has been proven that an optimized path can be found between point by connecting other resources or location which user needs to visit along the path. Also it helps giving user options of other possible paths between source and destination to choose from.

CHAPTER – 6

CONCLUSION

AND

FUTURE SCOPE

6.1 CONCLUSION

Optimal path planning is very essential for navigation of various computing devices such as robots, autonomous vehicles, transportation vehicles etc. All these computing devices can easily work and move on their own if they have the ability to plan the path and travel. In order for fast travelling, without collision even in complex as well as simple environment, a device should have a good navigating ability which can provide an optimal path to travel without any obstacles.

Many researches have been done in the past that involves path planning and optimization. Multiple algorithms are employed for efficient navigating of computing devices such as Dijkstra for vehicle routing, shortest path planning, emergency evacuation [4][8], particle swarm optimization (PCO) combine with simulated annealing for finding shorter and smoother paths in complex environment [6], Bee algorithm for game navigation [1], genetic algorithm for movement of robots [13] and many more. The techniques of path finding has advanced from classical path planning techniques to heuristic search techniques. Now the combination of these are used to further improve the efficiency.

Same is the case of navigation of vehicles in the city. Nowadays, metropolitan cities have web like framework or roadways where in order to reach a particular destination there are multiple routes available. But finding the best among those physically is not an easy task, especially for those who are new to the city. Hence, in this situation it becomes a need for advance navigation algorithm which is capable of fulfilling the traveler need and provide a best route based on those needs.

This dissertation aims at the same thing. Its purpose is to not only provide and efficient path but also a capability to navigate through the roads along with the resources / locations which the user wants to visit before reaching the destination.

Along with this, the proposed algorithm is also capable of giving user choices of all possible paths from the source to the destination.

This task is realized with the help of an algorithm which is made by combining the multiple path finding ability between two nodes provided by the Floyd – Warshall algorithm and the ability of 0/1 knapsack to prioritize resources / locations need to be visited before reaching the location as per the users choice. The accuracy of the algorithm when compared with other optimization algorithms, as shown through a comparative graph (Graph 5.4), in the previous chapter is 83%. This shows that the algorithm is capable of providing a path solution considering all major factors which are time, distance, traffic, fuel and resources.

6.2 FUTURE WORK

In future this algorithm can be implemented by creating an app for path navigation. It will be connected with the users details and the users can access this app through a secret key provided to them at the time of registration. Further this algorithm can be improved by adding the feature of fuel consumption when following the suggested path. With the help of satellites and GPS the traffic prediction feature can also be incorporated.

Work can also be done to reduce the complexity of the algorithm, in order to generate faster results. All these features and suggestions can help further enhance and improve the efficiency of the algorithm and the overall working of the system, which in turn can provide more optimal and user friendly results.

REFERENCES

[1] Aimi Najwa Sabri , Nor Haizan Mohamed Radzi, Azurah A Samah.(2018) “A study on Bee Algorithm and A* Algorithm for Pathfinding in Games”. IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 224-229.

[2] Mina G. Sadek, Amr E. Mohamed, Ahmed M. El-Garhy.(2018)” Augmenting Multi-Objective Genetic Algorithm and Dynamic Programming for Online Coverage Path Planning”. 13th International Conference on Computer Engineering and Systems (ICCES),475-480.

[3] Hironori Hiraishi.(2017) ”Route-planning based on a passenger condition for self-driving vehicles”. IEEE 16th Int'l Conf. on cognitive Informatics & cognitive computing , 329-334.

[4] Emna Mejri, Sousso Kelouwani, Yves Dube´, Omar Trigui, Kodjo Agbossou.(2017) ” Energy efficient path planning for low speed autonomous electric vehicle”, IEEE Vehicle Power and Propulsion Conference (VPPC).

[5] Hu Pan, Chen Guo, Zhaodong Wang. (2017) “ Research for Path Planning Based on Improved Astart Algorithm”. 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS). 225-230.

[6] Zhibin Nie, Xiaobing Yang, Shihong Gao, Yan Zheng, Jianhui Wang, Zhanshan Wang.(2016) “Research on Autonomous Moving Robot Path Planning Based on Improved Particle Swarm Optimization”. IEEE Congress on Evolutionary Computation (CEC), 2532-2536.

[7] E. E. Ogheneovo, E. Seetam.(2016) “A Heuristic Graph-Based Shortest Path Algorithm for Optimizing Routing Problem”.American Journal of Engineering Research (AJER,) 239-246.

[8] Yi-zhou Chen, Shi-fei Shen, Tao Chen, Rui Yang.(2014) “Path Optimization Study for Vehicles EvacuationBased on Dijkstra algorithm”. Procedia Engineering 71, 159 – 165.

[9] Wang Yu-qin, Yu Xiao-peng.(2012) “Research for the Robot Path Planning Control Strategy Based on the Immune Particle Swarm Optimization Algorithm” International Conference on Intelligent Systems Design and Engineering Application,724-727.

[10] Jinhao, Lu Chi Dong.(2012) “Research of Shortest Path Algorithm Based on the Data Structure” IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS),108-110.

[11] Bo Huang, Weiming Wu, and Yuning Zheng. (2011) “The Research and Improvement of Path Optimization in Vehicle Navigation” D. Jin and S. Lin (Eds.): Advances in MSEC Vol. 2, AISC 129, pp. 203–207.

[12] LI Guang-ru, ZHi Sun.(2010) “The Research on the Path Optimization of Container Truck Based on Ant Colony Algorithm and MAS”, International Conference on Web Information Systems and Mining,370-375.

[13] Low, C.Y., Chong, K. H., Salleh, K., Johnny, K.S.P (2010).” Path Optimization Using Genetic Algorithm Evolution”. IEEE Student Conference on Research and Development (SCOReD), 252-255.

[14] Tianmiao Wang, Bin Wang, Hongxing Wei, Yunan Cao, Meng Wang, Zili Shao(2008).” Staying-Alive and Energy-Efficient Path Planning for Mobile Robots”.American Control Conference 868-873

[15] Geethu Elizebeth Mathew(2015).” Direction Based Heuristic For Pathfinding In Video Games” Procedia Computer Science 47 ,262 – 271.

[16] Jose A. Mocholi, Javier Jaen, Alejandro Catala(2010).” An Emotionally Biased Ant Colony Algorithm for Pathfinding in Games”.Expert Systems with Applications: An International Journal.

[17] Zhou yefu, Li zaiyue(2010).” Research on logistics delivery route optimization based on Genetic Algorithm”. International Conference on E-Business and E-Government, 3407-3410.

[18] Björnsson, Yngvi;Vadim Bulitko ; Nathan Sturtevant. TBA*: Time-Bounded A*. Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09);2009,431-436.

[19] Zhu, Weiping, et al. "Waypoint graph based fast pathfinding in dynamic environment." International Journal of Distributed Sensor Networks 11.8 (2015): 238727.

[20] F. Zhan, Three fastest shortest path algorithms on real road networks: Data structures and procedures, Journal of Geographic Information and Decision Analysis, vol. 1, no. 1, pp. 70-82, 1997.3

[21] Cowley, Ben, et al. "Toward an understanding of flow in video games." Computers in Entertainment (CIE) 6.2 (2008): 20.

[22] Frana, Phil (August 2010). "An Interview with Edsger W. Dijkstra". Communications of the ACM. 53 (8): 41–47.

[23] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs" (PDF). Numerische Mathematik. 1: 269–271.

[24] Q. Sun, J. H. Shen, J. Z. Gu, An improved Dijkstra algorithm [J]. Computer Engineering and Applications, vol. 3 ,2002, pp.99-101.

[25] Mathew, Geethu Elizebeth. "Direction based heuristic for pathfinding in video games." Procedia Computer Science 47 (2015): 262-271.

[26] Saleh Alija, Amani. Analysis of Dijkstra's and A* algorithm to find the shortest path. Diss. Universiti Tun Hussein Onn Malaysia, 2015.

[27] Mocholi, Jose A., et al. "An emotionally biased ant colony algorithm for pathfinding in games." Expert Systems with Applications 37.7 (2010): 4921-4927.

[28] Algfoor, Zeyad Abd, Mohd Shahrizal Sunar, and Hoshang Kolivand. "A comprehensive study on pathfinding techniques for robotics and video games." International Journal of Computer Games Technology 2015 (2015):7.

[29] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics. 4 (2): 100–107.

[30] Klein, Dan; Manning, Christopher D. (2003). A* parsing: fast exact Viterbi parse selection. Proc. NAACL-HLT.

[31] Kagan E. and Ben-Gal I. (2014). "A Group-Testing Algorithm with Online Informational Learning" IIE Transactions, 46:2, 164-184.

[32] Vi Tran Ngoc Nha, Soufiene Djahel and John Murphy (2012).”A Comparative Study of Vehicles' Routing Algorithms for Route Planning in Smart Cities”. IEEE VTM 2012, Satellite Workshop of IFIP Wireless Days.

[33] Mohd. Nayab Zafar and J.C. Mohnta (2018) “Methedology for Path Planning and Optimization of Mobile Robots : A Review” International Conference on Robotics and Smart Manufacturing (RoSMa2018), Procedia Computer Science 133, 141 – 152.

[34] P.K. Mohanty and D.R. Parhi. (2014) “Journal of Experimental & Theoretical Optimal path planning for a mobile robot using cuckoo search algorithm.” December:37–41.

[35] Ouch W, Korczak J, "Optimization and global minimization methodssuitable for neural networks ". KMK UMK Technical Report 1/99; submitted to Neural Computing Surveys.

[36] Sadeghi, Javad; Sadeghi, Saeid; Niaki, Seyed Taghi Akhavan (10 July 2014). "Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm". Information Sciences. 272: 126–144.

[37] Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press.p.2

[38] "Understanding Denial-of-Service Attacks". US-CERT. 6 February 2013. Retrieved 26 May 2016.

[39] Taghavi Zargar, Saman (November 2013). "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks" (PDF). IEEE COMMUNICATIONS SURVEYS & TUTORIALS. pp. 2046–2069. Retrieved 2014-03-07.

[40] Khalifeh, Soltanian, Mohammad Reza (2015-11-10). Theoretical and experimental methods for defending against DDoS attacks. Amiri, Iraj Sadegh, 1977-. Waltham, MA. ISBN 978-0128053997.

[41] Goodin, Dan (28 September 2016). "Record-breaking DDoS reportedly delivered by >145k hacked cameras". Ars Technica. Archived from the original on 2 October 2016.

[42] Khandelwal, Swati (26 September 2016). "World's largest 1 Tbps DDoS Attack launched from 152,000 hacked Smart Devices". The Hacker News. Archived from the original on 30 September 2016.

[43] Kumar, Bhattacharyya, Dhruba; Kalita, Jugal Kumar (2016-04-27). DDoS attacks : evolution, detection, prevention, reaction, and tolerance. Boca Raton, FL. ISBN 9781498729659.

[44] "Imperva, Global DDoS Threat Landscape, 2019 Report" (PDF). Imperva.com. Imperva. Retrieved 4 May 2020

PLAGIARISM CHECK REPORT

ABSTRACT

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	296	Date	August 06,2020
Characters	1865	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	13 Unique Sentences

Content Checked For Plagiarism

ABSTRACT An efficient path optimization is said to find the shortest path from source to destination that takes less time, avoiding heavy traffic, where time being considered as the most important factor. But such is not always the situation. There has been a lot of research done on finding the optimized routes for many purposes like movement of autonomous vehicles, robot movement, aircraft movement, vehicle navigation etc. All these past researches focuses on finding a single efficient path based on a single predefined factor such as time or energy consumption. Most of the past approaches of path optimization had used algorithms like dijkstra, A* and Genetic algorithm. In this dissertation, work has been done to improve the ability of already existing researches on path optimization. It aimed at developing a more efficient path optimization algorithm that helped to overcome the major drawbacks of the previous system. The objective was not only to focus on the three factors namely time, distance and traffic but to include availability of resources / locations along the path to visit according to the

CHAPTER 1

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	927	Date	July 23,2020
Characters	5615	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	45 Unique Sentences

Content Checked For Plagiarism

1.1 PATH PLANNING AND OPTIMIZATION: Path planning is also called Motion planning. It is a computational problem that can be solved automatically by the computer, following a sequence of valid constraints which allows movement of the objects from the source to destination. For example, consider a navigating car in a city to a distant waypoint. It should find a path while avoiding footpaths, buildings, unconstructed areas etc. A path planning algorithm would take all these description and more as input, before providing instructions of a path suitable for movement of the car within a city to reach its destination It can be divided into two main categories; (i) local path planning and (ii) global path planning. In local path planning, the computational device is well aware about the area to be navigated as it is very limited and thus the destinations can be

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	848	Date	July 23,2020
Characters	5300	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	39 Unique Sentences

Content Checked For Plagiarism

Thus path planning and optimization plays the major role in various other domains such as vehicle routing, which is movement of the vehicle within a city, Artificial intelligence, which is mostly used in gaming for the movement of various characters, logistics movement, for the movement of automated vehicles, emergency evacuation, it is important during the time of disaster or an accident, delivery drones or autonomous drone, for finding a correct path to reach the destination safely and many others [19]. All these problems have encouraged many researchers to focus on effective path planning and optimization techniques. 1.4 VARIOUS APPLICATIONS OF PATH PLANNING AND OPTIMIZATION: Optimal Path finding plays vital roles in many domains like vehicle routing, robot path planning, logistics movement, Artificial Intelligence (AI), games, emergency evacuation and various others [19]. In terms of vehicle route optimization it is necessary to have a shortest path following all factors of optimal solution. Looking for a shortest path between two locations in the road network is a very challenging task in vehicle routing, distribution and logistics industry [32]. And in many transportation application that involve

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	923	Date	July 23,2020
Characters	5771	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	46 Unique Sentences

Content Checked For Plagiarism

1.6.2 A* ALGORITHM In the year 1967 Peter Hart, Nils Nilsson and Bertram Raphael proposed an informed search algorithm called the A* search algorithm [25]. It is extensively used in the fields like computer science and pathfinding for problem solving [1]. The creation of A* algorithm was a part of "the shakey project" having the aim of building a mobile robot capable of planning its own actions. It is also a breath first search algorithm as it is devised in terms of weighted graphs. It starts from a particular node and aims at finding that path from the source to destination which have minimum distance, shortest time and smallest cost. Being seen as n extension of Edsger Dijkstra's 1959 algorithm A* is capable of achieving better performance by using heuristics as its guide in search. What makes this algorithm more special is that it has "brains". This means that it is

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	179	Date	July 23,2020
Characters	1128	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	9 Unique Sentences

Content Checked For Plagiarism

1.7 DISSERTATION OUTLINE The organization of rest of the dissertation is as follows: Chapter 2 In this chapter, security challenges and issues path planning and optimization approach is discussed. Objectives of this chapter are to explain different types of attacks on cloud and GPS in detail. Chapter 3 In this chapter, there are reviews from various national and international journals and publications. It is done to identify the real problem statement for doing appropriate research. Chapter 4 In this chapter, the proposed work is discussed and explained in detail for finding optimal path on the specific parameters and factors. Chapter 5 In this chapter the metrics that were used to measure the performance of proposed work along with diagrams that illustrate the performance measurements, the implementation details and results of the detection

CHAPTER 2

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	984	Date	August 06,2020
Characters	5998	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	44 Unique Sentences

Content Checked For Plagiarism

There are various security issues that occurs when creating a project that involves huge amount of data processing, various different parameters and constraints, use of networks such as GPS, user authentication etc. Some of the security issues that can occur in the proposed work are as follows: 2.1 DENIAL OF SERVICES Denial-of-Service attack(Figure 2.1) is a cyber attack intended to hamper the normal functioning or completely shutting of a machine, device or a network preventing its intended users (i.e. members, office employees, account holder etc.) to access it. The main aim of this attack is to overload the target with the redundant requests, send in abundance, or sending any information that triggers a crash, which cause the system to shut down completely depriving all the requests which were legitimate from being completed [38]. This attacks targets

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	930	Date	August 06,2020
Characters	5635	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	44 Unique Sentences

Content Checked For Plagiarism

2.1.6 HOW CAN ONE TELL IF A SYSTEM/NETWORK IS UNDER DoS ATTACK ? The are various indications that can tell whether a particular system is under denial of service attack or not. They are: The files will take longer time to load, due to the slow network performance. Sudden loss of connection to a network. Unresponsiveness of a particular website, when loaded on web page. Increase in processing time. Unnatural behavior of the system. In the proposed work this attack is can be prevented because of the following reasons: Each client using the system has to be registered using his/her personal information. When registered the client is provided with the secret key through which he/she can access the system. Hence it cannot be accessed by any other unauthorized user. The secret key provided to the user is used to access the location of

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	408	Date	August 06,2020
Characters	2427	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	17 Unique Sentences

Content Checked For Plagiarism

 All the security administration is done in a single place, thus there is no need for manual configurations of security or security updates. It is reliable as with the right security measures, which are centralized, the users can use data and applications on cloud very safely. Various cloud security services provide live monitoring of data and hence improve its availability As the technologies are growing more and more organizations can easily rely on cloud and cloud security. It allows them to easily use up resources along with security, in minimum cost. Figure 2.12: Benefits of cloud security 2.2.4 CLOUD SECURITY CHALLENGES There are various security challenges when come to securing of public cloud as it is being managed by the third party and is used through internet. Some of these challenged are: The IT teams should have the capability of

CHAPTER 3

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	992	Date	July 24,2020
Characters	6157	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	48 Unique Sentences

Content Checked For Plagiarism

3.1 LITERATURE SUMMARY The advancement in path planning technologies in order to provide an optimal path has put human efforts in navigation at rest. Due to rapid economical growth and changing working conditions, the time on people's hand has reduced immensely. Thus for a newbie in a city or person going outside the city by road, wasting time in prior path planning is futile. There has been many researches in the past and are still going on intensively in the field of optimized energy efficient path planning for various purposes like vehicle routing, disaster management, routing of autonomous vehicles, robot movement in games and otherwise, UAV movement etc. All these researches show various algorithms and techniques for an efficient path planning. This chapter further highlights those techniques, the comparison among them and their involvement

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	830	Date	July 24,2020
Characters	5268	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	39 Unique Sentences

Content Checked For Plagiarism

E. E. Ogheneovo and E. Seetam (2016) [7], aim was to determine the shortest and the most cost-effective routes. This research used dijkstra algorithm to find the shortest routes while and heuristic search graphs are used to find a cost-effective solution. The proposed algorithm focused in developing a graph based algorithm for solving the problem of route optimization especially in Nigeria tourism. It provides a model focused on synthesis of heuristic graph and adaptive routing. When compared to the other routing algorithms the proposed method showed more better results in efficiency. Yi-zhou Chen et.al (2014) [8], aim was to develop a dynamic road network model for vehicles evacuation during emergency events, like, earthquakes, hurricanes, fires, terror attacks, accidents etc. All these events may lead to loss of life and health of people. The proposed research used dijkstra algorithm that provides optimal evacuation path in three different cases namely morning peak, common and evening peak. The results produces an efficient theoretical basis and a well predictive method for optimal path selection for emergency evacuation and emergency rescue decision in public places mostly where population density is

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	969	Date	July 24,2020
Characters	6154	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	43 Unique Sentences

Content Checked For Plagiarism

Geethu Elizebeth Mathew(2015) [15], proposes a new method that is capable of generating a higher quality path using less time and memory as compared to other existing solutions in video games. It also reviews all the existing widely used pathfinding solutions. The author has introduced a new heuristic method for increasing the speed of pathfinding process on a map represented using grids with uniform cost. The most important part of this approach is the identification of the nodes to be explored and proper usage of the data. As a result this direction based heuristic approach enhance the speed and the efficiency while reducing the memory consumption. This is because the number of node to be explored are limited and are carefully selected. The significance of this approach is in the video games. Jose A. Mocholi et.al. (2010) [16], aim is to find a method capable of taking into account the emotions of the game characters in path finding. This is done because the games these days are more visually realistic graphics which increases the realism of the characters in the game. The Ant colony Optimization (ACO) algorithm is used for decision making that also takes into account the emotions needed to give a more

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	620	Date	July 24,2020
Characters	3928	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	25 Unique Sentences

Content Checked For Plagiarism

11. Aims at providing an improved route optimization by improving dijkstra algorithm. Dijkstra Algorithm The results showed great improvement in the search time as it saves half of the time for every search. Bo Huang et.al. (2011) [11] 12. Aims at providing an intelligent system that takes in real-time traffic information through GPS and solves the problem scheduling container trucks Ant Colony Algorithm This model is capable of providing real -time dynamic scheduling scheme to solve the problem of dynamic doc scheduling. LI Guang-ru and ZHi Sun (2010)[12] 13. The aim was to method capable of optimizing the path of gantry robots and providing more efficient and economic movement. Genetic Algorithm The result of this research was very positive as the motor turn on and the robot travelling time were shortened. Low, C.Y. et.al (2010) [13] 14. The authors proposed two approaches to solve the staying-alive and energy-efficient path planning problem Greedy Travelling Salesman

CHAPTER 4

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	993	Date	August 06,2020
Characters	5815	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	45 Unique Sentences

Content Checked For Plagiarism

4.1 PROPOSED WORK The use of applications like Google maps gives an individual a shortest path with lesser time and traffic. However such systems provide only a single route from the source to destination without giving any further choices that could help a user plan out a path according to its resources and necessities. But such is not always the situation. Some time the path planning is more driven towards the resources available on that path or the fuel in the vehicle. A real time example of

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	975	Date	August 06,2020
Characters	5389	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	45 Unique Sentences

Content Checked For Plagiarism

 (Figure 4.4) Figure 4.4: The values of all the elements of A1 matrix are filled using A0 matrix The above step is repeated for all the other vertices in the graph. 1. For vertex k = 2, matrix A2 is created. Since value of k is 2, thus the second row and second column elements are left as it is and the diagonal elements are filled as 0. Then using matrix A1, step 3 is repeated to create matrix A2. (Figure 4.5) Figure 4.5: The values of all the elements of A2 matrix are filled using A1 matrix 2. For vertex k = 3, matrix A3 is created. Since value of k is 3, thus the third row and third column elements are left as it is and the diagonal elements are filled as 0. Then using matrix A2, step 3 is repeated to create matrix A3. (Figure 4.6) Figure 4.6: The values of all the elements of A3 matrix are filled using A2 matrix 3. For vertex k = 4, matrix A4 is created. Since value of k is 4, thus the fourth row and fourth

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	662	Date	August 06,2020
Characters	3502	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	23 Unique Sentences

Content Checked For Plagiarism

Figure 4.10: Flowchart explaining work module 4.5.2 ALGORITHM USED: 1) let distance be D[n] x [n] and value be V[n] x [n] 2) for matrix of distance initialized to minimum distance initialized or ∞ and value to maximum value gaining. 3) for each edge (u, v) do 4) D [u][v] ← distance 5) for each vector v do 6) V [u][v] ← Value 7) end for 8) end for 9) for m from 1 to v 10) for x from 1 to v 11) for y from 1 to v 12) if D[x][y] > D[x][m] + D[m][y] 13) D[x][y] ← D[x][m] + D[m][y] 14) Value ← V[x][m] + V[m][y] 15) Pr ← Knapsack(V, x, y) 16) end if 17) end for 18) end for 19) end for Knapsack (V, x, y) 20) for w from 0 to x do 21) c[0][w] ← 0 22) end for 23) for i from 1 to n do 24) c[i][0] ← 0 25) end for 26) for i from 1 to w do 27) if wi ≤ value then do 28) V ← value 29) else 30) V ← value[i+1] 31) end if 32) return V 4.5.3 DESCRIPTION OF THE ALGORITHM: In the proposed algorithm the aim is find an optimal path based on the resources and choices entered by the user. The user will be able to see all the paths available from the entered source to the destination, also will be given the most optimal path which use less distance, time fuel etc. The

CHAPTER 5

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	967	Date	August 06,2020
Characters	5385	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	39 Unique Sentences

Content Checked For Plagiarism

5.1 RESULT ANALYSIS: The algorithm mentioned above was implemented in various scenarios. The results and conclusions were derived: A 10 km busy road was taken as an input for the navigation. This was done when there were no specific resources or locations needed to be accessed along the path. The road was navigated at different time period between 8:00 am (morning) to 12:00 am (night) and the time taken to travel that busy road at different time period was recorded. The details are shown in Graph 5.1, according to which, the minimum time taken to navigate the road is in the morning between 8:00 am to 10:00 am. During this time period i.e. morning, the traffic is very less even on the busy road. Thus it took only 14.5 minutes to reach the destination. Whereas the most time taken to travel the same road was during the time period between 5:00 pm to 8:00 pm, i.e. it took 35.6 minutes. The same 10 km busy road was taken as input for navigation and this time the resource / location was chosen to visit along the path. Graph 5.2, shows the time recoded for the same time periods as Graph 5.1, when the road was travelled with resources. According to the graph, it shows that the minimum time taken even when the

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	526	Date	August 06,2020
Characters	3353	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	25 Unique Sentences

Content Checked For Plagiarism

Lastly the above task of navigation was implemented using various algorithms, with purpose finding the most fasible algorithm to provide optimal result. The accuracy of all the algorithms were recorded as shown in graph 5.4 and table 5.3. According to the graph and the table, when implementing Floyd-Warshall algorithm it showed 72% accuracy in providing multiple path from the source to destination. Rest of the algorithms that are dijkstra, prism and krushal were 67%, 56% and 58% accurate. Among which prisms has the lowest accuracy. Even though Floyd –Warshall was 72% accurate and is capable of giving multiple paths between source and destination but is still not enough to provide optimal results, where the traveler need other resources/locations to access along the path. Hence, an algorithm was proposed which combines the features of Floyd – Warshall, which is capable of providing multiple paths from source to destination and 0/1 knapsack, which is capable of choosing resources based on priority and maximum value. The algorithm showed 83% accuracy when implemented to find a all possible paths from source to destination along with the choice of most optimal path which covers all the resource along

CHAPTER 6

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	557	Date	August 06,2020
Characters	3451	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	24 Unique Sentences

Content Checked For Plagiarism

6.1 CONCLUSION Optimal path planning is very essential for navigation of various computing devices such as robots, autonomous vehicles, transportation vehicles etc. All these computing devices can easily work and move on their own if they have the ability to plan the path and travel. In order for fast travelling, without collision even in complex as well as simple environment, a device should have a good navigating ability which can provide an optimal path to travel without any obstacles. Many researches have been done in the past that involves path planning and optimization. Multiple algorithms are employed for efficient navigating of computing devices such as Dijkstra for vehicle routing, shortest path planning, emergency evacuation [4][8], particle swarm optimization (PCO) combine with simulated annealing for finding shorter and smoother paths in complex environment [6], Bee algorithm for game navigation [1], genetic algorithm for movement of robots [13] and many more. The techniques of path finding has advanced from classical path planning techniques to heuristic search techniques. Now the combination of these are used to further improve the efficiency. Same is the case of navigation of

PLAGIARISM CHECK FOR ENTIRE DESSERTATION BY TURNITIN

Dissertation

ORIGINALITY REPORT

12%	4%	7%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PUBLICATION FROM THIS WORK

1) **“Path optimization: A Potential Approach to Path Planning using Machine Learning”** has been accepted in the International Conference on Recent Trends in Electrical and Computer Science Engineering (ICEECS - 2020) held at Uma Nath Singh Institute of Engineering and Technology, Purvanchal University, Jaipur and published in Scopus Indexed Journal.

2) **“A Potential Approach to Path Planning and Optimization for Vehicle Routing”** has been accepted by Proteus Journal, which is multidisciplinary journal and is indexed in Web of Science group and UGC approved group – II.