International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Swarm Intelligence Based Algorithm for Efficient Routing in VANET

Gagan Deep Singh, Manish Prateek, Hanumat Sastry G

Abstract*: Many recent researchers are working to optimize solutions in the field of Vehicular Adhoc Network. However, none of them has yet claimed that it will fulfill all the challenges of such a dynamic region. VANET in itself is a complete area of study, research and improvements. Most of the researchers and industry consortiums has given their hypothesis and solution that depends on their predefined scenarios but no complete solution has designed until yet. Through this research work, the authors concluded that bioinspired solutions can be used to integrate along with VANET for a much accurate and optimized solution. The performance of VANET depends on various scenarios and due to the unpredictable behavior of the vehicle movement, no concrete solution can be claimed as of now. We incorporated Swarm Intelligence in VANET through the Ant Colony Optimization algorithm and found that the performance of VANET has enhanced by avoiding the entire congested path as it senses the pheromone trail. We have implemented and tested the results using open source software like Instant Veins, Simulation of Urban MObility (SUMO) and MObility model generator for VEhicular networks (MOVE). SUMO has used for testing the traffic simulation and MOVE is used to design model. Python for the script. The OSM used to take a map of Dehradun city. When we performed the experimental setup and found that the result confirms in reducing the traveling time of the nodes, which makes nodes faster and managed even it helps in saving the hydrocarbon fuels. During our approach, we have devised our own algorithm that has improvised the present Ant Colony Optimization algorithm and has concluded that the average traveling time of the nodes minimized through our approach.

***Keywords* : Ant Colony Optimization, Dehradun, OSM, Swarm Intelligence, VANET.**

I. INTRODUCTION

Every year many lives lost in road accidents. Technology can be a boon that can minimize causalities during transport. The intelligent transport system is one of the keys to it and many of the worldwide consortiums are already working for it. By using Vehicular Adhoc Network (VANET) along with artificial intelligence, the best and optimized algorithm has applied and it results in the solution of the real-time VANET problems. In this paper, we came up with a new approach in which Swarm Intelligence through Ant Colony Optimization incorporated in VANET model and we found the revolutionary results. However, this is a proposed hypothetical solution and many other parameters are need to be taken care before it can apply in the real-time environment for VANET solutions. Vehicular Ad hoc Network (VANET) uses wireless communications to communicate among themselves and other vehicles just like that of Mobile Ad hoc Network (MANET) [1]. MANET works on Optimized Link State Routing (OLSR) protocol but the characteristics of OLSR not fit in resources like energy consumption and other hardware [2]. It takes data packets from source to destination while traveling in OLSR protocol. But if we consider the same for VANET then QoS decreases during routing while applying the OLSR protocol in the predefined model and scenarios of Urban mobility environments. Hence, to get the best routing process we need to revise the configuration setup of OLSR to meet the features of VANET. We can reconfigure the parameters like bandwidth, delay, routing network to optimize QoS in VANET using OLSR [3]. Due to the unpredictable nature of vehicle and driver's driving habits, it is almost impossible to apply MANET solutions for it as this will result in routing and link failures in an environment like Urban traffic conditions. Such aspects make more challenging for eager researchers to develop best suitable, optimized and efficient routing protocols for VANET. In VANET vehicle nodes uses the wireless communication system to disseminate the information among vehicle-to-vehicle, vehicle-to-infrastructure, and infrastructure-to-infrastructure communications [4].

Generally, vehicle node mobility patterns are cannot be determined as they move to depend on the traffic scenarios, lights, road or highway structures, and driving behavior and experience of the person seated at the driver's seat [5,6] also came up with many security challenges suggested need cryptographic solutions for VANET. These days all the vehicles are equipped with the features of Global Positioning System, and other equipped devices/sensors that support to provide various information such as signal timings, traffic estimation, fuel consumption, routing decisions best path, shortest route etc. [7]. If VANET improvised with intelligence then it may put the ability that can enhance road safety, fuel efficiency and even comfort for the drivers as well as for commuters [8]. Intelligent VANET has made an entirely new area of research for the Intelligent Transport System to design new, automate and smart transportation systems [9].

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Gagan Deep Singh*, School of Computer Science, University of Petroleum & Energy Studies, Dehradun, India. Email: gagan@ddn.upes.ac.in
Dr. Manish Prateek, School of Computer Science, University of Petroleum & Energy Studies, Dehradun, India..
Dr. Hanumat Sastry G, School of Computer Science, University of Petroleum & Energy Studies, Dehradun, India

It is becoming new era for making cities smart in traffic management. [10]. VANET makes users to decrease the traveling and waiting time. However, traveling time cane be reduced by opting for the shortest paths. So, if the roads bandwidth is not enough to cope up with the increased random numbers of vehicle of these shortest path then obviously there will congestion occurs like that of Ants behaviour as shown in Fig. 1 and 2. From this, we can clearly conclude that opting the shortest path is not always results in the fastest route [11]. Therefore, to overcome such situations we will opt the lengthy route to avoid traffic of nodes/vehicles and lesser traffic signals to minimize stoppages at red signals. Hence, we designed and tested our own algorithm that works by applying Ant Colony Optimization technique. After reviewing various bioinspired algorithms, we came to know that ACO work in a more optimized way as the behavior of Ants movement and their movement pattern is is best suited for VANET scenarios. Therefore, in our research, we deployed our methodology using swarm intelligence from implementing our algorithm and named as *"ManishHanumatGaganUPES Algorithm (MHGU)"*.

We have tested MHGU algorithm on different scenarios found that the vehicle nodes facing lesser congestion and hence decreasing the travel time and waiting time. In this manner MHGU also playing role in reducing the pollution and at same time lowering the fuel consumptions. We also found the optimum results as now vehicles not wasting time at red signals. Similarly, it MHGU algorithm reduced the congestion on the routes. We have considered the vehicle nodes are fit and are properly traveling on the route without any breakdowns. The roads are also in best possible conditions without any civil work. These both points taken as default for all the scenarios in our Hypothesis.

In previous researches, we have found that an ant colony optimization algorithm was modified as they have chosen the route that has minimum pheromone comparatively to the maximum [12]. This results in avoiding the congestion on the route that has followed by ants. Similarly, suggested preemptive adaptive algorithm for reducing waiting period on traffic signals, which, reduces average queue length on junctions and intersections [13]. In our new devised MHGU algorithm, we have enhanced the Modified Ant Colony Optimized algorithm using Swarm Intelligence that made it preemptive in nature it is able to avoid congestions and reduced the time of waiting at signals. The outcome of a new proposed algorithm is the optimized solution in VANET challenges.

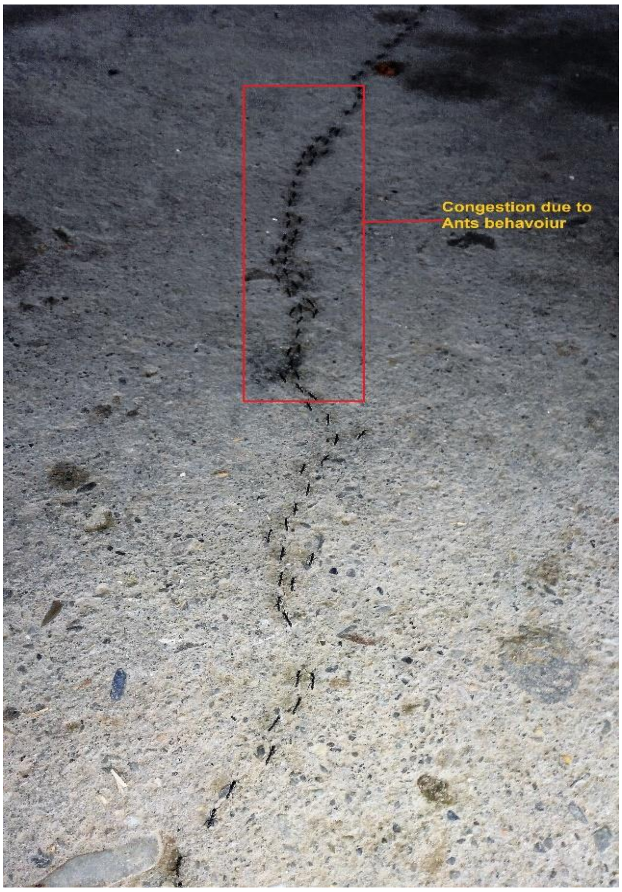

Fig. 1. Ant behaviour (photo taken on 02-06-2018)

Fig. 2. Zoom Visual of Congestion

II. REVIEW OF PREVIOUS WORK DONE

This has shown in many reviews that reducing the time of waiting of the vehicles is only be achieved if we are able to provide a free movement of vehicles in a designed route or at the traffic signals if they can be optimized. Nevertheless, optimization and free movement of vehicles will not possible because we do not have checks on the input of vehicle nodes during real-time traffic.

However, the tuning of traffic signals has already presented to deal with random traffic during signals in the research of many researchers in 2014 [14]. Many updates has come now for traffic signaling and enhanced it for congestions and traffic in 2013 and 2015 [15], [16]. Generally, the programmed traffic signals fixed as per their timing parameters and this cause the congestion of the vehicles because we cannot predict the number of vehicles at any of the directions [17]. Now, recently new sensors are playing a role changing game for it, which can be installed at the roadsides and thus random increase and decrease of the vehicle nodes can be predicted or determined through these detectors. Hence, traffic signal timings can adjusted as per that input [18].

However, we have reviewed and concluded that mostly the cyclic adaptive approach has done mainly. However, there is a probability that at the time of the green signal no vehicle is available for the movement. This results in congestion at the other directions where many vehicles have stopped at a red signal and waiting for green signal. On the other side, the signal is green but no vehicle is available to pass. At, these scenarios preemptive methodology is best suited, so that green signal can be executed at the lane wherever we find an increase in the congestion of the vehicles. Routing of the nodes helps vehicles to find the destination point and reach in lesser time. In VANET, no routing technique has yet finalized because of nondeterministic movement of the vehicles. Similarly, in 2014 [19] proposed that optimum routes can be predicted through dynamic routing through the information gained from the equipment like sensors and detectors. Two factors that are positioning and topology are the parameters through which the routing algorithm can categorized. Dijkstra's algorithm is one of the best routing algorithms, based on positioning and is able to find the shortest path in any of the two nodes in a network [20]. Whereas, reactive, proactive and hybrid are algorithm that is based on topology-based algorithms [21]. A heuristic search technique based on Greedy Best First Algorithm and suited for routing in VANETs [22]. Heuristic approach used to provide the solution in its best suitable time through estimation of the routing problem. This may not be the most optimized solution though it can achieve the aim rapidly [23]. In comparison to the Dijkstra's algorithm, it does not select the current node. However, the next closest node is selected in every other move. Hence, this algorithm returns the best route as all hops are looking for the further nodes.

We also reviewed Swarm behavior and found that swarms have the prompted ability to resolve complex and real-time challenges that are not possible to get the answer through previously reviewed algorithms. Hence, swarm intelligence is best suited for real-time scenarios of traffic simulations. One such Artificial Intelligence Swarm Algorithm is an Ant Colony Optimization (ACO) approach. Due to its nature, ACO is best suited for designing the routing in network domains and in communication paths. This ACO algorithm shows the ability that can develop the self-learning and managing situations to act in resolving the routing of real-time problems in VANET [24], [25]. ACO works similarly as that of Ants behavior. Ants left a chemical called pheromone on the entire path while traveling through their route. In addition, while other ants travel through these paths, then ants sense pheromone to get the updated information of the route to follow the same path [26]. Ant Colony is termed for the group of ants, which are traveling in search of food or relocation themselves. Sometimes, the group of these ants discovers accumulated pheromone. When they found a higher concentration of this chemical then only they follow that path, otherwise, discard. As, pheromone is a chemical hence it also evaporates when the route is not followed by the ants for a longer duration [27]. We have also seen many other domains adopted ant colony technique for optimizing other types of problem using Traveling Salesman Problem [28]. Furthermore, ACO modified to get the newer version for the VANET traffic scenarios in complex routing paths of the vehicles to get more optimized results [29]. The main objective of ACO in VANET is to minimize the stoppage durations during traveling, which can result in reducing the total time taken for journey, irrespective of the route length opted by the nodes [30].

A modified Ant Colony Optimization algorithm for minimizing the congestion on the routes chosen by vehicles. They modified the pheromone behavior, which can deviate the nodes from their selected path that avoids congestion. The new modified ACO algorithm tested for accuracy and effectiveness by performing experiments on real-time Delhi University road network, simple road network and complex road network [31]. These researchers not included the traffic signal that comes in the way and that may increase the average travel time due to stoppages at signals. Therefore, in our present research outcome, the integration performed for the modified Ant Colony Optimized in preemptive traffic algorithm. That gave very innovative results by reducing the average travel time at congestions. To calibrate our proposed newly devised algorithm we have compared it with Dijkstra's and modified Ant Colony Optimized algorithm. The results obtained through our experiments show a reduction in average travel time and minimized the waiting time in VANET.

III. METHODOLOGY FOLLOWED

We focused only on easily available software for implementation of experiments. The best part is they all are open source software.

1. Open Street Map
2. Instant VEINS 4.7.1
3. MObility model generator for Vehicular networks(MOVE)
4. OMNET++ 5.3
5. Python
6. Traffic Control Interface(TraCI)

We implemented this on HP workstation and allocated dedicated 2 processors and 4 GB of memory to the Virtual machine allocated for instant Veins 4.7.1. We also recommend others to use minimum 4GB of RAM for such experimental setups. We have adopted the following methodology for deploying the test shown in Fig. 3.

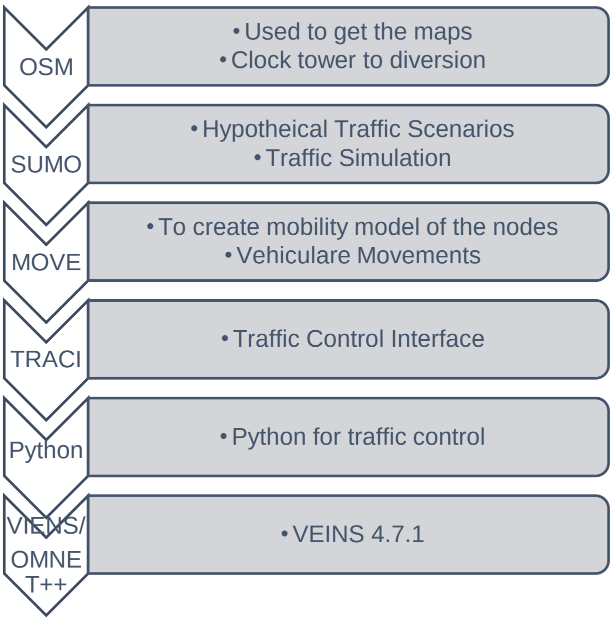

Fig. 3. Methodology Followed for simulation Test

IV. WORKING ON ALGORITHM

In our hypothetical model, we have assumed that generally, the vehicles have to wait on its path because of the traffic signals at the junctions or sometimes has to wait during congestion created by civil works, breakdowns, accidents, blockage of roads and many more. Thus, we thought that to minimize overall duration of journey we need to reduce the time wasted due to signals and unwanted stoppages. Therefore, we came up with the new MHGU algorithm that is capable of reducing the total travel time. In our hypothesis, we have considered that the all the roads are in proper working condition along with traffic signals.

V. MHGU (MANSH HANUMAT GAGAN UPES) ALGORITHM

We observed, that mostly a traveler waste his/her time while waiting during the traffic signals or spends time during congestions caused by any of the reason like civil works, breakdown of other vehicles, etc. Therefore, we proposed a MHGU algorithm in our research that is capable of reducing the time for waiting at traffic signals as well as optimizes the path through congestion avoidance during the journey. Hence, it can also help in reducing the traveling cost as optimization results in reducing consumption of the hydrocarbon fuels and decreases the driver's irritation due to congestions and traffics.

Presently, we have observed that the major cause of congestion is due to the large duration of stoppages of the vehicles at traffic signals while waiting for their turn to pass the intersections. They wait because of a non-balanced timed signal or some predefined time actuated signals through a program to manage the time but without considering the real-time actual vehicle cluster. Generally, it is a mind setup of the humans to promptly, select for the shortest path for their destination. This results in an increase in the congestion as all the drivers are willing to drive on the same chosen path. Till yet, researchers has only focused either in optimizing the travel time by reducing the journey duration or working on routing algorithms in VANET. Hence, in our proposed MHGU algorithm we have taken care of both of the parameters and tried to optimize them together for designing the efficient routing algorithm in VANET. In our proposed algorithm, we have finely tuned and improvised the modified ACO algorithm by integrating it with preemptive techniques of the traffic signal at intersections, which may come during the entire journey. Implementation of MHGU algorithm results in minimized travel time through optimized non-congested path and lesser waiting time at traffic signals. This also makes to reach a destination even cheaper as less fuel burns.

A. The Proposed MHGU Algorithm

In an optimized ACO algorithm, a vehicle may across many or few signal points. Those traffic signal points are the major cause that can increase the stoppage durations during the travel. Hence, our objective is here to reduce the waiting time at traffic signals first. MHGU algorithm has strengthen with preemptive traffic signal algorithm [32] and is here, used to decrease overall journey duration to minimize the waiting time at the traffic signals. In our methodology to achieve our objective, firstly we have chosen a path with lesser pheromone chemical, as it has minimum pheromone hence it will be also less congested. Now, whenever traffic signals interrupted on this path the preemptive algorithm executed that reduces the stoppage duration at the traffic signals. In our methodology, we have assumed the route from a source node to a destination node. This path is nothing but the roads in VANET. Where different roads intersects each other is the intersection of roads along with traffic signals on it. Here, each road is having same three lanes as are in real-time road conditions. The first lane is for the left way, the second road designed to go straightway and the third will be used to go in right through its right turn. To test and implement we took a network simulation in which vehicles have random starting point and random destination points. Now, when the vehicles start moving randomly, it shows that their initial positions are fluctuating consequently as of adjacency matrix.

As we have applied swarm intelligence in our approach, so the movement of vehicles will be like that of ants on its journey. Firstly, the vehicle or the node gathers information about its initial position and moves further on the road. It also senses the value of the pheromone chemical to reach its desired destination. This could be also possible that the vehicle is in between somewhere on the road and so we can take its present status as the initial position. If we found that, the value of destination position is the same as that of current position this means either vehicle is stable now or it has reached its destination, then the next vehicle is considered and repeating so on, until all the vehicles reached their desired destinations. Now, when the present position of the node is in between the vehicles or waiting due to some congestions or red signal of the traffic light at intersections, then we allocate the next node as the new position to that node moving on the same lane. Else, route having lesser pheromone chemical value and minimum nodes than the assumed threshold will be chosen, from the current position of the next adjacent vehicle.

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Now, after confirmation that the vehicle is verified, whether it is waiting at a traffic signal and then make it moved to some other road with less congestion that can take it to its destination. Now, the selected road is analyzed for any of the signals across it, if found put preemptive algorithm on it. Else, we will update the recent position of the vehicle to the position of the next node for that vehicle. Therefore, as per the ACO principle, pheromone chemical value will rise up for the opted road and non-selected roads will show a decrease in the pheromone chemical value.

Our next target is to determine the congestion on the roads. For this, the value of the threshold is to be compared and considered with the total number of vehicles taken during simulation tests. Here, threshold need to calculate through experiments and then it takes for analyzing the congestion. Whenever we found traffic signals, at the route of modified ACO algorithm, we applied preemptive algorithm and the outcome is in decreasing the waiting time even more at the traffic signal stoppages.

We here devised our MHGU algorithm with similar to that of Ant Colony Optimization algorithm with our own integration of preemptive traffic signaling for better outcome. We assumed that the entire route is fully functional for vehicles without any civil work and breakdowns. As it is, described below:

- Take the vehicle as nodes of the network path. (same will be considered for simulation experiments)
- When the vehicle identified in between the nodes then select the next node on that path and make it the next node.
- Else, another node having minimum pheromone chemical value to be selected.
- If, traffic signals are present in between the path of the current node and the next node, then apply the preemptive algorithm.
- After applying the preemptive algorithm, update the position of the vehicle nodes and find the next node.

B. The Proposed Framework

The framework that we designed to opt for our research methodology shown in Fig. 4.

We started our work systematically to achieve the outcome of our hypothesis. The Process flow diagram of the experiments performed are as shown below in Fig. 5.

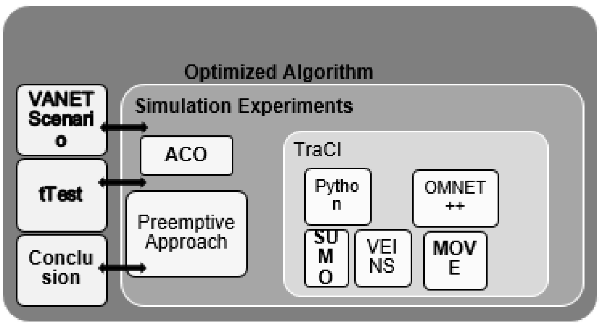

Fig. 4. Designed and adopted framework

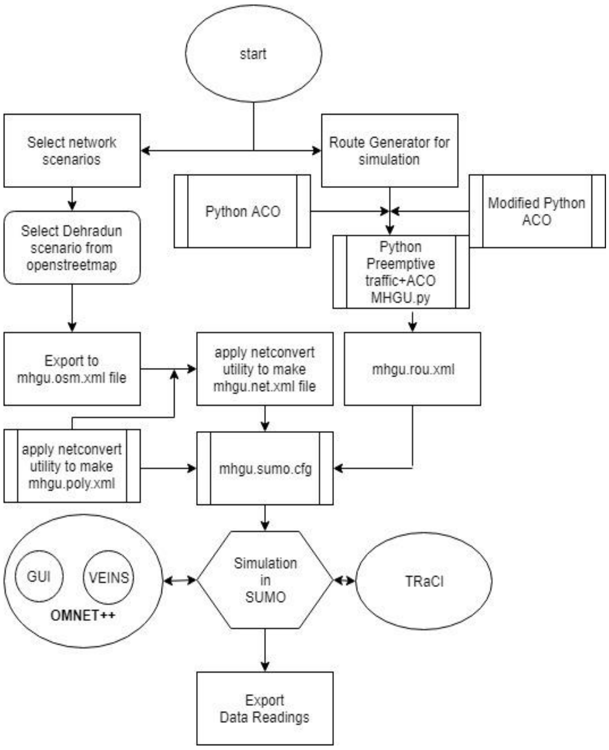

Fig. 5. Process Flow Diagram of the experiments performed

We have developed our own MHGU algorithm by integration of two distinguished algorithms. MHGU algorithm was able optimize routing in VANET as described in Fig. 6. GitHub (https://github.com/takuya-ki/swarm-intelligence) [33] used to for python script and modified as per our problem statement. Executed three scenarios for VANET simulation, which also includes Dehradun clock tower to diversion as in Fig. 7. of www.openstreetmap.org [34]. Deployed Instant VEINS 4.7.1 on HP Workstation with 4GB of RAM to run the simulation. The statistics of the executed simulation was then converted into excel format for compiling results and graphs. The gathered data then computed for t-test to confirm the validity of this hypothesis. The result of t-test concludes that our hypothesis is correct and our MHGU algorithm reduced the time to wait of the vehicles that also results in lesser travel time. Hence, here we can state that as per our hypothesis confirms MHGU algorithm is responsible for efficient routing in VANET.

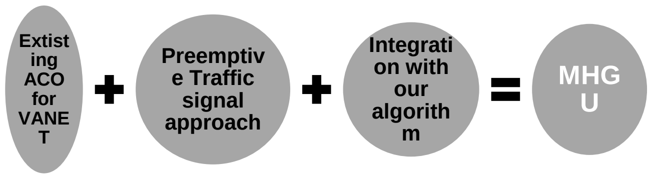

Fig. 6. Optimization Process

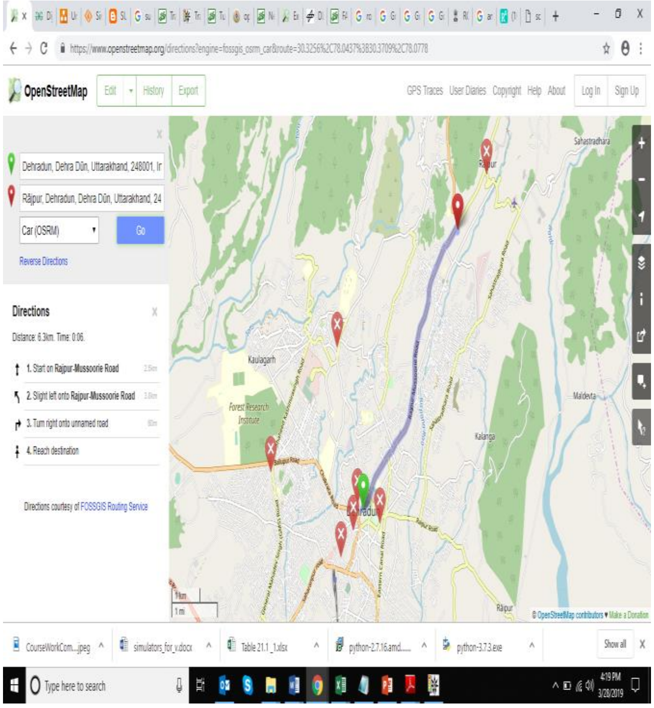

Fig. 7. Dehradun map from openstreet.org

Ant-based routing builds the routing path and is able to select the optimal routing path for communication among nodes. Pheromone value assigned to each route that can vary along with time duration. As there are constraints of road topology, VANET's network topology updated dynamically and new routes established for communication in order to select fresh routes. If, Pheromone decreases it may lead to broken links and may cause vehicles to adopt lengthy paths. Thus, optimum route discovery could not initiated using a single node; multiple nodes are required to explore unfamiliar briefest ways among the longest way. We applied our modified script of ant-based mobility aware routing through the AntWMNet routing protocol for testing and comparing the benefits of ACO.

According to our modified ant-based routing, initial Pheromone value is set for each vehicle in the network and a fading factor is also, initialized to avoid routes having less Pheromone called dump routes. Whenever vehicles move, if there is any update in routing information, Pheromone updates and depends upon collected feedback, current routes may ignored. High-speed vehicles can frequently update the Pheromone value and current route may be congested, as other Ants may adopt the same route due to its high Pheromone value to avoid this situation, Pheromone is managed in such a way that no route can have the Pheromone more than allowed limit i.e. threshold. To keep the Pheromone alive, it is updated using route feedback, if it is positive, Pheromone is refreshed otherwise Pheromone is faded out. If vehicles are moving randomly, extra time is required to update the faded Pheromone as compared to alive Pheromone.

C. Swarm Intelligence Ant Colony Optimization Methodology adopted

We have applied self-organizing principles of insect societies for coordinating the population of artificial agents, which collaborate to solve computational problems of VANET routing through experiments using simulation tools. In our case, we have used Instant Veins 4.7.1.

The ants use pheromone trail that keeps long-lasting memory for the whole process of ant search and ants update this by themselves.

ACO Algorithm

Next Hop → r
Source Node → S
Destination → D
Routing Probability → Rp
Pheromone_Fading_Factor → pFf

Response of the Ants

Pheromone_Feedback1
{
Init_Fbk := iFbk; //initial Feedback
Pv_Fbk := pFbk; //+ve Feedback
ng_Fbk := ngFbk; //-ve Feedback
}

Pheromone Value

Pheromone_Val
{
Init_Pheromone := P; //initial Pheromone Value
MAX_Pheromone := mxP; //Max. Pheromone
Min_Pheromone := mnP; //Min Pheromone
Refresh_Factor_Pheromone := rfP; // Pheromone update interval
}

Velocity

Velocity
{
MAX_v := mxV; //Maximum velocity
Min_v := mnV; //Minimum velocity
Acceleration_Factor_Pheromone :=afP; //change in Pheromone due to mobility }

Initial route establishment

Step1: initialize(Source,Destination)
Step2: for each node Ni
{
Initialize(Anti,Ni,P);//initial node
}
Step3: build_route()
{
Get_next(P);
If(Ni,r,S,P)
(Rp++,r,n,destination d);
}

Route Management

if(pFbk)
{
if(P<mxP)
{
P++;
afP = rfP;
}
}
else
{
P--;
afP--;
}
if(P<mnP)

```
        {
        Dump_route(Ni,P);
        }
        if(afP != rfP)
{
pFf++;
}
        if(pFf)
{
                Dump_route(Ni,P);
                }
```

Proposed Optimized Algorithm

Our devised MHGU algorithm is only valid if all roads are in the best usage condition and along with running traffic light signals. The MHGU algorithm as summarized below:

Algorithm: VANET Optimization using Ant Colonization Optimization

Input: S_1= Network1 (Simple Road), S_2 = Network2 (Complex Road), S_3 = Network3 (Dehradun Clock Tower till Mussoorie Diversion Road), Dehradun, India taken through www.openstreetmap.org, *D* = Dijkstra's Algorithm, P_{ACO} = Simple Ant Colony Optimization, MP_{ACO} = Modified Ant Colony Optimization, Pre_{RTA} = Preemptive Road Traffic Algorithm, Ћ ∊Threshold Value, $\mathcal{N}$= Number of iteration
Output: $\Re$ (Optimized result over VANET after applying ant colonization)

1. Set value of S_1, S_2, S_3 for VANET simulation criteria
2. Set common parameters mentioned in Table 1, for all the three scenarios S_1, S_2, S_3 and implementing D, P_{ACO}, and MP_{ACO} correspondingly
3. for $i = 0$: $\mathcal{N}$
4. $S_1, S_2, S_3 \leftarrow D$
5. $\Re_1 \leftarrow Result$
6. $S_1, S_2, S_3 \leftarrow P_{ACO}$
7. $\Re_2 \leftarrow Result$
8. $S_1, S_2, S_3 \leftarrow MP_{ACO}$
9. $\Re_3 \leftarrow Result$
10. *if congestion = = high && congestion >* Ћ
11. $S_1, S_2, S_3 \leftarrow MP_{ACO}$
12. $\Re_4 \leftarrow Result$
13. *then*
14. *continue*
15. $\Re \leftarrow \Re_1, \Re_2, \Re_3, \Re_4$

return $\Re$

VI. EXPERIMENTAL PROCESS AND RESULTS

We have used Veins, https://veins.car2x.org/download/ [35] ova image as a framework with OMNET++ simulator for simulation of traffic scenario as per the details shown in Table 1. We applied ant colony routing protocol AntWMNet/AODV that, from the library. For faster processing only 30 vehicle nodes selected in simulation parameter. To run simulation we selected speed of 20ms, 40ms, and 80ms consecutively.

A. Simulation Analysis

Table I: Simulation Parameters

Parameters	Configuration
Routing	AntWMNet/AODV protocol
Area	1500 X 1500
Nodes	30
Speed	20ms, 40ms, 80ms
Packets	100bytes
Protocol	MAC 802.11p
Traffic	CBR
Sampling interval	2.5ms
Simulation duration	Varies on system performance
Network Simulator	Instant Veins 4.7.1/OMNET++
Scenarios	Road Network1, Road Network2, Exported OSM Dehradun

B. Experiments Performed to test our MHGU Algorithm

We used Instant Veins 4.7.1 that has OMNET++ and SUMO. The local client-server model is established using TraCI(TRaffic Control Interface) that comes with SUMO in src folder. SUMO and OMNET++ connected by a middleware TraCI. In Veins, sumo-launchd.py is a script executed for a client-server connection of SUMO and TraCl via OMNET++. After initiating the simulation process OMNET++ requests to TRaCI server to receive the location of nodes, through this a real-time road network generated on OMNET++ Simulation [36]. Same method applied for our defined parameters in omnetpp.ini file. We ran these simulations for our scenarios as per hypothesis. MAC 802.11p protocol deployed for communicating among vehicles/nodes.

There are some of the 802.11p specific parameters for Network Interface cards like Transmission Power, carrier frequency, bitRate etc. were also used. The TraCI specific settings are sending our request to localhost and on port number 9999 where SUMO TraCI server is already started and Mobility parameters. All this is taken care by OMNET++ and VEINS together and finally complete simulation and automatic rerouting of vehicles on sensing congestion threshold.

From this simulation experiments, we can conclude that demonstration of the modelling and simulation of real-world proposed traffic scenarios and scheduled traffic congestion, which acts as an event generator for all following vehicles, vehicles receiving signals again rebroadcasts the event information to other vehicles in the vicinity. The algorithm automatically changes the route of following vehicles once the congestion threshold detects and threshold time is lapsed.

In our methodology, we have taken simple road network, complex road network and OSM of Dehradun road network in Instant Veins. To calculate the efficacy of newly devised algorithm MHGU the various scenarios simulated in defined parameters and to control the simulation we used Python scripts that integrated with MHGU algorithm. The simulation experiments used to test and propose our new algorithm for an efficient routing and dissemination in VANETs. [37].

We executed the simulation repeatedly using different vehicle/node density to get the average value for various scenarios. Mainly vehicles follow the path defined by Dijkstra's algorithm.

However, MHGU algorithm is able to define the non-congested route by applying modified Ant Colony optimization principles. Whenever the vehicles face traffic signal then it compares for the threshold value, if value exceeds then MHGU algorithm is applied. Through these vehicles again starts to move through the different available route having least congestion. Hence, avoiding the waiting time and long stoppages.

Python used for scripting the algorithm. Simulation scenarios tested for three different road networks as discussed before in this paper. Then travel time and waiting time of the nodes recorded in a table for comparison. After comparison, this clearly seen from the results that our devised MHGU algorithm is better as it results in reducing the travel as well as waiting time of the vehicles.

Table 2 to 5 shows the data recorded through MHGU algorithm applied on various scenarios. Fig. 8 to 15 represents the graphs of simulation tests. When vehicles wait at stoppages, they burn fuel and increase the pollution in the environment. Hence, through MHGU algorithm we can play a vital role in decreasing the fuel consumption of the vehicles as well as decrease their participation in polluting the environment.

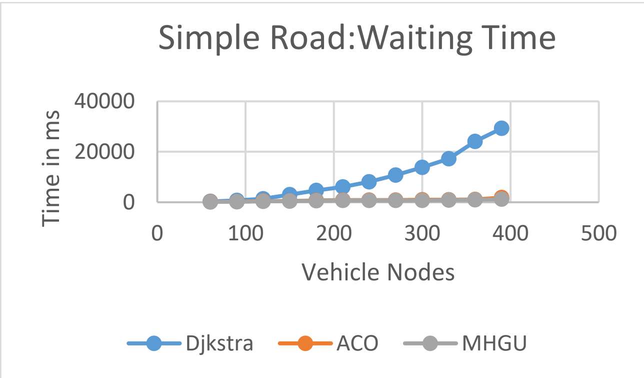

Fig. 8. Waiting Time plot on simple road scenario

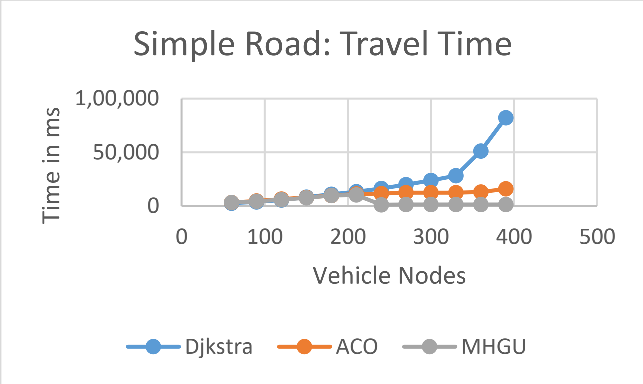

Fig. 9. Travel time plot on simple road scenario

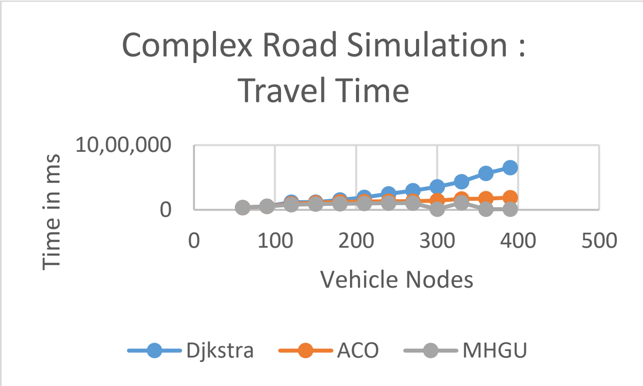

Fig. 10. Travel Time plot for Complex Road scenario

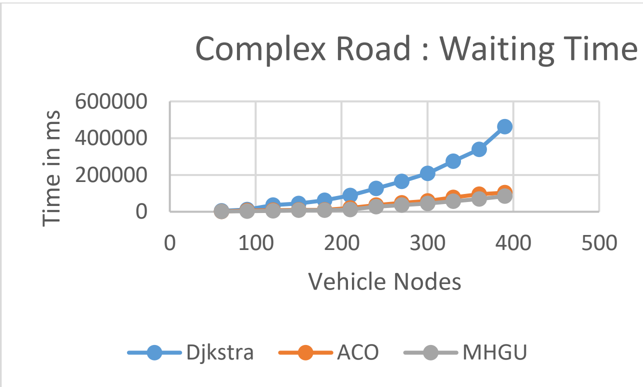

Fig. 11. Waiting Time plot for complex road scenario

Table II : Waiting and travel time of non-complex road

Vehicle Nodes	Waiting Time			Travel Time		
	non-complex road simulation					
	Dijkstra's	ACO	MHGU	Dijkstra's	ACO	MHGU
60	283	157	132	2,437	3,077	2,845
90	702	210	198	3,648	4,491	4,125
120	1,411	439	365	5,365	6,155	5,521
150	3,018	534	448	8,070	7,755	7,541
180	4,720	777	652	10,676	9,639	9,478
210	6,088	813	702	13,005	11,121	9,985
240	8,073	845	726	15,980	11,265	1,005
270	10,754	887	741	19,685	12,140	1,065

300	13,757	938	786	23,692	12,276	1,085
330	17,267	969	824	28,186	12,281	1,096
360	24,087	1,101	950	50,916	12,688	1,102
390	29,284	1,825	1,201	82,157	15,684	1,115

Table III: Waiting time for complex road

Vehicle Nodes	Waiting Time			Travel Time		
	Complex road simulation					
	Dijkstra's	ACO	MHGU	Dijkstra's	ACO	MHGU
60	5018	1620	1582	35,188	35,635	34,954
90	12,174	5742	3214	55,516	55,230	52,419
120	36,538	7432	4568	116326	95,094	78,546
150	44,597	9240	7542	117838	110,861	88,672
180	62,629	10,001	8659	151291	122,942	92,457
210	89,117	20,617	12,405	192514	126,836	99,875
240	126963	36,529	28,035	244723	131,334	101,458
270	165376	48,079	36,547	297464	135,107	105,478
300	207776	59,270	45,025	354539	143,862	105,987
330	274113	78,108	56,457	434487	169,021	110,259
360	338134	95,566	69,542	562966	171,275	111,231
390	462546	103,635	84,579	651297	188,726	111,598

Table IV: Waiting Time and Travel time for openstreetmap.org scenario

Vehicle Nodes	Waiting Time			Travel Time		
	Dehradun road simulation					
	Dijkstra's	ACO	MHGU	Dijkstra's	ACO	MHGU
60	2878	529	342	46,789	42,967	40,215
90	10,975	1371	1125	75,277	46,630	42,541
120	21,624	3101	2541	107,066	101,207	65,489
150	36,361	7286	3627	140,261	135,160	85,476
180	45,967	9487	5487	173,127	170,757	96,587
210	59,248	12,209	7825	208,321	205,078	108,541
240	78,841	17,264	8964	249,799	222,477	112,547
270	94,210	22,005	9247	287,198	238,850	146,547
300	120,114	23,922	9876	335,756	282,950	152,598
330	140,041	36,941	10,054	377,792	294,444	179,876
360	177,282	44,251	10,265	436,553	382,338	185,478
390	209,047	50,279	11,429	490,118	399,071	216,547

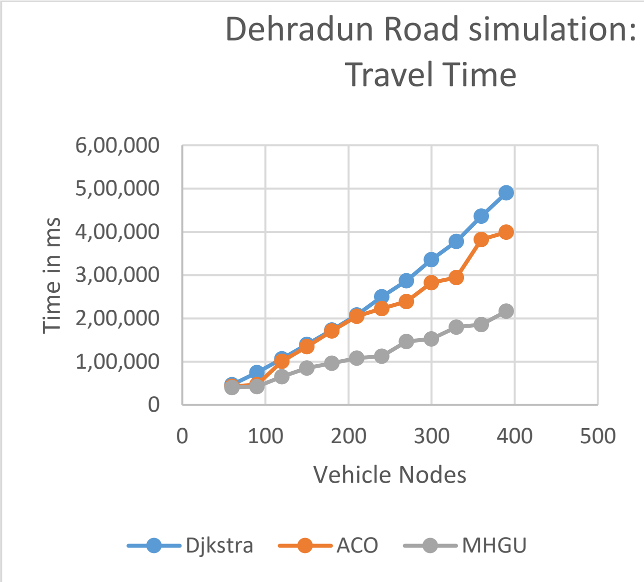

Fig. 128. Travel Time plot for Dehradun Road Scenario

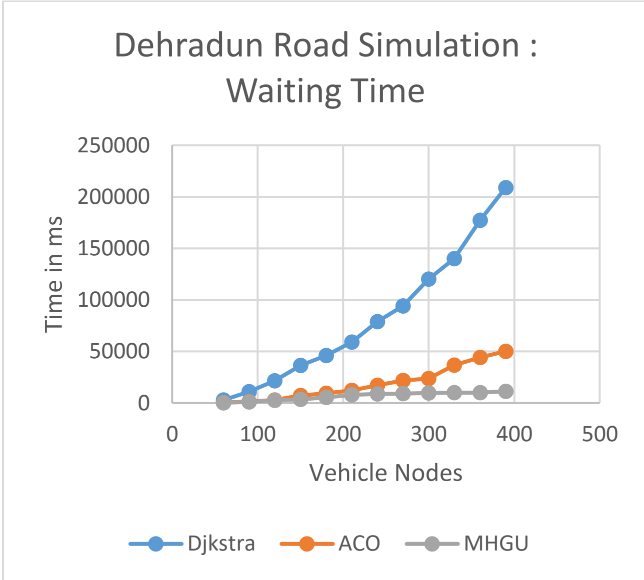

Fig. 139. Waiting Time plot of Dehradun Road simulation

Table V: Clock Tower to Mussoorie Diversion Time

Vehicle Nodes	Waiting Time			Travel Time		
	Clock Tower to Diversion road simulation					
	Dijkstra's	ACO	MHGU	Dijkstra's	ACO	MHGU
60	18,618	1600	1325	90,051	74,162	65,478
90	49,856	3729	1987	150,720	101,590	79,543
120	61,750	7490	2354	198,286	143,385	85,479
150	120676	11,677	4529	279,660	175,432	93,157
180	158252	15,051	6587	365,911	228,537	101,598
210	243077	48,684	7896	486,826	241,382	106,587
240	389057	58,004	9648	667,761	289,037	112,689
270	670521	69,547	10,254	983,066	405,814	126,926
300	1,057,375	79,428	10,658	1,603,810	425,784	148,756
330	1,131,682	88,751	11,025	2,413,606	454,785	178,542
360	1,201,634	96,547	11,542	3,418,917	485,478	198,755
390	1,692,484	131,524	25,487	4,545,012	504,125	254,139

Retrieval Number: E2857039520/2020©BEIESP
DOI: 10.35940/ijitee.E2857.039520
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

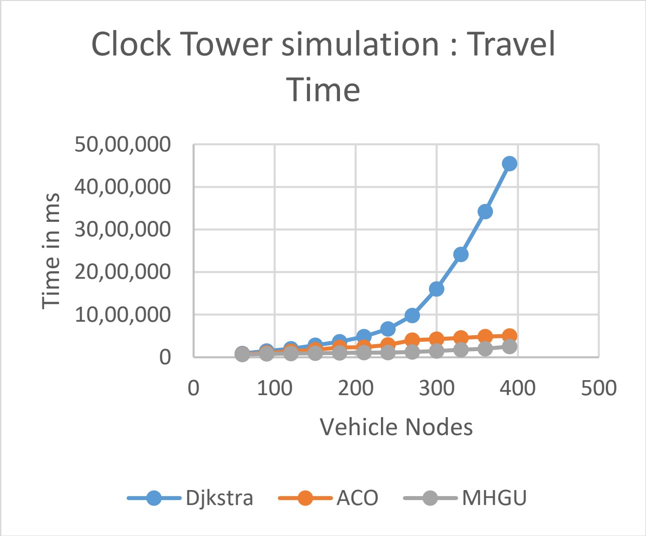

Fig. 14. Clock Tower travel time simulation plot of Dehradun

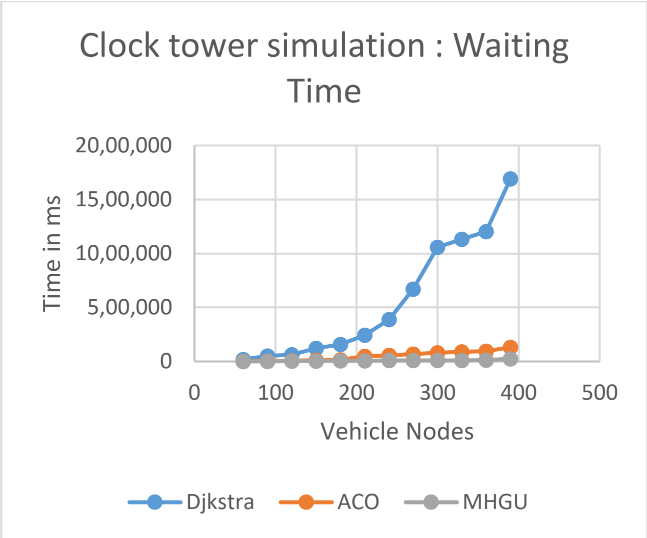

Fig. 15. Waiting time plot of Clock Tower to Diversion simulation

C. Testing of Hypothesis

For analyzing the performance of the proposed MHGU algorithm, we have used recognized statistical technique t-test. t-test performed for confirming our hypothesis through the sample results generated in simulations tests. The t-test used to verify the acceptance of our specified hypothesis. We performed this using two paired samples. Null hypothesis (H_0) and alternate hypothesis (H_1) defined as follows:

$$H_0 : \mu_1 \leq \mu_2:$$

and

$$H_1 : \mu_1 < \mu_2:$$

where μ_1 is taken as overall travel time calculated from Dijkstra's algorithm and μ_2 is taken as overall travel time calculated using MHGU algorithm. t-test statistic is performed to evaluate whether the null hypothesis is valid or not. Equation (1) performed to calculate the t-test. Where n_1 and n_2 the sample sizes of the two algorithms, μ_1 and μ_2 are the means of two algorithms, and s_1 and s_2 are the standard deviations.

We used online tools for getting the result of our t-test https://www.graphpad.com/quickcalcs/ttest1.cfm

Equation 1: t-test

$$t = \frac{(\sum D)/N}{\sqrt{\dfrac{\sum D^2 - \left(\dfrac{(\sum D)^2}{N}\right)}{(N-1)(N)}}}$$

Algorithm to calculate ttest:

Step1: Take *Sample1* as *X* and *Sample2* as *Y*

Step2: Subtract each *Y (Sample1)* from each *X (Sample2)*

Step3: Sum of all the value calculated from Step2 $\sum D$

Step4: Square the difference calculated in Step2 $(X-Y)^2$

Step5: Sum of all the squared differences calculated in **Step4:** $\sum D2$

Step6: Apply Formula as

$$t = \frac{(\sum D)/N}{\sqrt{\dfrac{\sum D^2 - \left(\dfrac{(\sum D)^2}{N}\right)}{(N-1)(N)}}}$$

Step7: To get the *degree of freedom* subtract 1 from all sample size.

Step8: Find *p-value* in the t-table using the degree of freedom in Step7; if specified *alpha level* is not there.

Then use *0.05 i.e. 5%*. (For this sample problem)

Step9: Compare t-table value from Step8 to our calculated value. The calculated t-value > the table value at an alpha level *0.05*. If, the *p-value* is less than the alpha level: $p<0.05$. Then the null hypothesis rejected as no difference between means.

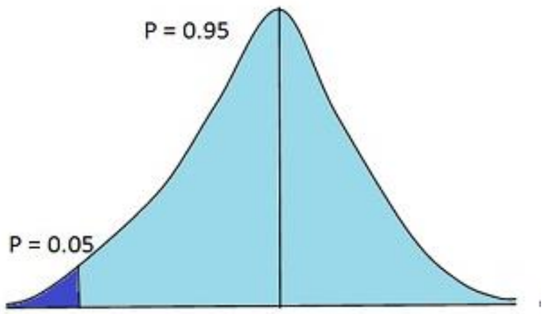

Fig. 16: One tail t-test

D. t-test implementation for proving our Hypothesis

The statistical t-test is perform to prove the efficiency of our devised MHGU. The t-test applied on the simulation test results of Dijkstra's, ACO algorithm and our MHGU algorithm. Two sample data of two algorithms taken at a time to compare them. The t-test applied on the simulation results recorded for overall travel time to hypothesis test for the validity of the null hypothesis. $\mu 1$ taken as the mean of travel times for the modified ACO algorithm and MHGU algorithm and $\mu 2$ taken as the mean of travel times for Dijkstra's algorithm and modified ACO algorithm.

We investigated t-test where $\alpha = 0.04$, 0.06, 0.11 and 0.15 as they are for different significance values. These tests performed for all the three scenarios in simulated environment of Dijkstra's, ACO and MHGU. The same repeated for multiple times to gather the record of sample data. The probability p taken here for acceptance or rejection of our hypothesis. Where $p < 0.04$, considered as rejection of null hypothesis and hence, we accept the alternate hypothesis as it has 96% confidence. Else, p was compared again with α value and reject the null hypothesis if, value is < 0.06 and hence, we accept the alternate hypothesis as it has 96% confidence. Similarly repeated for rest of the significance levels i.e. 0.11 and 0.15.

Therefore, from this we can conclude clearly the efficiency of MHGU algorithm. We have rejected the null hypothesis and with 96% confidence, we have accepted the alternate hypothesis. Furthermore, we can also conclude that ACO algorithm has even more efficient than Dijkstra's algorithm because with 96% confidence the alternate hypothesis accepted for ACO algorithm. Hence, from this we have proved that our proposed MHGU algorithm is best when compared with Dijkstra's and ACO algorithms. This is what was required as per our problem statement, i.e., devising an optimized routing algorithm in VANET.

VII. CONCLUSION AND FUTURE SCOPE

Our tested MHGU algorithm that is devised in our research and deployed using Instant Veins to reduce the total travel time of vehicles in VANET. The optimization achieved by reducing the stoppage duration and selecting the less congested route for the journey. The newly devised algorithm integrates the preemptive traffic light and ACO techniques to achieve our goal. Through this, MHGU algorithm has acquired the ability to find the optimized route. So that the nodes can avoid the congested path by applying the ant behaviour techniques.

The efficiency of our MHGU algorithm verified by comparing it with preexisting algorithms using simulation experiments and statistical tools. The outcome of our proposed algorithm is reduced total travel time and minimized stoppages of the vehicle nodes. This also observed that MHGU algorithm results best at the time of heavy congestion but shown negligible improvement when vehicle density was less. Through its ability to avoid congested path, it makes the journey faster and hence, plays a vital role in lowering the fuel consumption and reducing air pollutions produced by the vehicles.

This research can further extended to implement and test the same experiments on real time vehicle movements. Industry either can fund for this project or can apply the same methodology for their own tests. We hope in near future many more improvements and optimization is will raise to deploy this communication technology for information dissemination in VANET.

ACKNOWLEDGMENT

We wish to thank Mr. Jatender Sharma and Dharmendra Chauhan of IT Department, who facilitated us with the desired computing resources, without which it was not possible to perform simulation experiments without lagging and delays.

REFERENCES

1. P. Raikwar, and H. Gupta, "Network Congestion Minimization using Enhanced AOMDV Routing in MANET". International Journal Electrical Electronics and Computer Engineering,7(2) 2018, pp. 51-56.
2. C. Feng, Z. Li, S. Jiang, and R. Zhang. Data Aggregation and Routing Guidance with QoS Guarantee in VANETs. International Journal of Distributed Sensor Networks, 2014.
3. H. Alwan, and A. Agarwal. "Multi-objective QoS routing for wireless sensor networks". In Computing, Networking and Communications (ICNC), 2013 International Conference on (pp. 1074-1079). IEEE.
4. G. D. Singh, R. Tomar, G. H. Sastry, and M. Prateek. "A Review on VANET Routing Protocols and Wireless Standards", Smart Computing and Informatics, (pp 329 – 340), Springer, 29 Otcober2017.
5. S. Dhankhar, and S. Agrawal. VANETs: A survey on routing protocols and issues. International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), 2014, 3 (6):13427–35.
6. M. N. Mejri, J. B. Othman, and M. Hamid, "Survey on VANET Security Challenges and possible cryptographic solutions", Vehicular Communications, Elsevier, 2014 Vol. 1, pp. 53-66.
7. S. U. Rehman, M. A. Khan, T. A. Zia, and L. Zheng. Vehicular Ad-Hoc Networks (VANETs)-An overview and challenges. Journal of Wireless Networking and Communications. 2014 (3):29–38.
8. J. Kakarla, S. S. Sathya, B. G. Laxmi, and B. B. Ramesh. A Survey on Routing Protocols and its Issues in VANET. International Journal of Computer Applications. 2011. 28 (4):38–44. doi:10.5120/3373-4663.
9. U.S. Department of Transportation. 2004. Intelligent Transportation System (ITS). http:// www.its.dot.gov/index.htm (accessed January 25, 2014).
10. P. K. Singh, S. Shrama, S. K. Nandi, and S. Nandi, "Multipath TCP for V2I communication in SDN controlled small cell deployment of smart city", Vehicular Communication, Elsevier, Vol. 15, 1-15, 2019
11. Bell, J. E., and P. R. McMullen. (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics 18 (1):41–48. doi:10.1016/j. aei.2004.07.001.
12. V. Jindal, H. Dhankani, R. Garg, and P. Bedi. MACO: Modified ACO for reducing travel time in VANETs. Proceedings of the 3rd International Symposium on Women in Computing and Informatics (WCI-2015). India, ACM, 97–102
13. P. Bedi, V. Jindal, R. Garg, and H. Dhankani. A preemptive approach to reduce average queue length in VANETs. Proceedings of the 4th International Conference on Advances in Computing, Communications and Informatics (ICACCI'15), 2015. Kochi, India.

14. J. C. Dias, P. Machado, D. C. Silva, and P. H. Abreu. An inverted Ant Colony Optimization approach to traffic. Engineering Applications of Artificial Intelligence 2014.36:122–33.
15. J. C. Falcocchio, and H. S. Levinson. Road traffic congestion: A concise guide, 2015. vol. 7. Switzerland: Springer International Publishing.
16. Shivani, Shivani, and J. Singh. Route planning in vanet by comparitive study of algorithms. International Journal of Advanced Research in Computer Science and Software Engineering 2013. 3 (7):682–89.
17. R. P. Roess, E. S. Prassas, and W. R. McShane. Traffic engineering, 2011. 4th ed. New York, USA: Pearson Higher Education.
18. N. Maslekar, M. Boussedjra, J. Mouzna, and H. Labiod. VANET based adaptive traffic signal control. Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC Spring), 2011. Budapest, Hungary:IEEE, 1–5.
19. S. Rana, and K. C. Purohit. A review of various routing protocols in VANET. International Journal of Computer Applications 2014. 96 (18):28–35. doi:10.5120/16896-6946.
20. M. Altayeb, and I. Mahgoub. A survey of vehicular Ad hoc networks routing protocols. International Journal of Innovation and Applied Studies 2013. 3 (3):829–46.
21. A. Dua, N. Kumar, and S. Bawa. A systematic review on routing protocols for vehicular Ad Hoc networks. Vehicular Communications, Elsevier 2014. (1):33–52. doi:10.1016/j. vehcom.2014.01.001.
22. M. J. Piran, G. R. Murthy, and G. P. Babu. Vehicular Ad Hoc and sensor networks; principles and challenges. International Journal of Ad Hoc, Sensor and Ubiquitous Computing 2 (2):38–49. doi:10.5121/ijasuc.2011.2204.
23. P. Syal, and T. Kaur. A study of routing protocols for vehicular Ad-Hoc networks. International Journal of Engineering Trends and Technology.2014. 15 (1):25–29. doi:10.14445/ 22315381/IJETT-V15P206.
24. M. Dorigo, G. D. Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. 1999. *Artificial Life* 5 (2):137–72. doi:10.1162/106454699568728.
25. S. Geetha, P. T. Vanathi, and G. Poonthalir. Metaheuristic approach for the multi- depot vehicle routing problem. Applied Artificial Intelligence, 2012. 26 (9):878–901. doi:10.1080/ 08839514.2012.727344.
26. G. Cabanes, E. V. Wilgenburg, M. Beekman, and T. Latty. Ants build transportation networks that optimize cost and efficiency at the expense of robustness. Behavioral Ecology, 2015. 26 (1):1–9.
27. M. Dorigo, and S. Thomas. Ant colony optimization. Cambridge, Massachuswetts, USA: MIT Press. 2014.
28. W. Elloumi, H. E. Abed, A. Abrahama, and A. M. Alimi. A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. Applied Soft Computing 25:234–41. doi:10.1016/j.asoc.2014.09.031.
29. Y. Bin, Y., Y. Z. Zhen, and Y. Baozhen. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research. 2009., 196 (1):171–76. doi:10.1016/j. ejor.2008.02.028.
30. V. Jindal, H. Dhankani, R. Garg, and P. Bedi. MACO: Modified ACO for reducing travel time in VANETs. Proceedings of the 3rd International Symposium on Women in Computing and Informatics (WCI-2015). India, ACM, 97–102
31. G. D. Singh, M. Prateek, and G. H. Sastry, "Swarm Intelligence based efficient routing algorithm for platooning in VANET through Ant Colony Optimization", International Journal of Innovative Technology and Exploring Engineering, 2019. Vol. 8, Issue 9, 1238-1244
32. P. Bedi, V. Jindal, R. Garg, and H. Dhankani. A preemptive approach to reduce average queue length in VANETs. Proceedings of the 4th International Conference on Advances in Computing, Communications and Informatics (ICACCI'15), Kochi, India, IEEE, 2089–95.
33. https://github.com/takuya-ki/swarm-intelligence
34. https://www.openstreetmap.org
35. https://veins.car2x.org/download/
36. R. Tomar, G. D. Singh, M. Prateek, G.H. Sastry G. H. Modelling and Simulation of Vehicle to Vehicle Communication, International Journal of Control Theory and Applications, 2017. Vol. 10, No. 15, 2017.
37. M. Chahal, and S. Harit, Optimal path for data dissemination in Vehicular Ad hoc Networks using meta-heuristic. Computers and Electrical Engineering 76 (2019) 40-55.

AUTHORS PROFILE

Gagan Deep Singh is an Assistant Professor in School of Computer Science at University of Petroleum and Energy Studies Dehradun. He is postgraduate in Computer Science and pursuing Doctorate in the area of optimization of VANET routing and its performance. He has presented many research papers in national and international conferences and published research papers in international journals. He is having more than nine years of teaching and six years of system integration and administration experience. His area of interest is System integration, optimization and performance.

Dr. Manish Prateek is Professor and Dean in School of Computer Science at University of Petroleum and Energy Studies Dehradun. He is Doctorate in the field of Robotics. He is having a vast experience of twenty-three years and associated with many research consortiums. He has many of the publications in reputed national and international journals He has also chaired many national and international conferences. Under his guidance many scholars has completed their doctorate in the area of Computer Science, IT and Robotics. His area of expertise is in teaching, robotics, computers and Information Technology.

Dr. Hanumat Sastry G is an Associate Professor and Assistant Dean in School of Computer Science at University of Petroleum and Energy Studies Dehradun. He is Doctorate in Computer Science and have a rich experience of twelve years in teaching. He has also published many research papers in reputed journals. He is also guiding many doctorate research scholars. His area of expertise is in machine learning, e-learning, cloud computing and IT security.