

# Modeling Ant Colony Optimization for Multi-Agent based Intelligent Transportation System

Shamim Akhter<sup>1</sup>, Md. Nurul Ahsan<sup>2</sup>

Applied Intelligent System and Information Processing Lab  
Dept. of Computer Science and Engineering  
International University of Business Agriculture and  
Technology (IUBAT), Dhaka, Bangladesh

Shah Jafor Sadeek Quaderi<sup>3</sup>

Applied Computing,  
Faculty of Computer Science, and Information Technology  
University of Malaya  
Kuala Lumpur, Malaysia

**Abstract**—This paper focuses on Sumo Urban Mobility Simulation (SUMO) and real-time Traffic Management System (TMS) simulation for evaluation, management, and design of Intelligent Transportation Systems (ITS). Such simulations are expected to offer the prediction and on-the-fly feedback for better decision-making. In these regards, a new Intelligent Traffic Management System (ITMS) was proposed and implemented - where a path from source to destination was selected by Dijkstra algorithm, and the road segment weights were calculated using real-time analyses (Deep-Neuro-Fuzzy framework) of data collected from infrastructure systems, mobile, distributed technologies, and socially-build systems. We aim to simulate the ITMS in pragmatic style with micro traffic, open-source traffic simulation model (SUMO), and discuss the challenges related to modeling and simulation for ITMS. Also, we expose a new model- Ant Colony Optimization (ACO) in SUMO tool to support a multi-agent-based collaborative decision-making environment for ITMS. Beside we evaluate ACO model performance with exiting built-in optimum route-finding SUMO models (Contraction Hierarchies Wrapper) -CHWrapper, A-star (A\*), and Dijkstra) for optimum route choice. The results highlight that ACO performs better than other algorithms.

**Keywords**—Intelligent Traffic Management System (ITMS); Simulation of Urban Mobility (SUMO); traffic simulation; Contraction Hierarchies Wrapper (CHWrapper); Dijkstra; A-star (A\*); Deep-Neuro-Fuzzy Classification

## I. INTRODUCTION

A new intelligence traffic management system (ITMS) [8] [11] [20] [21] [22] [30] was proposed and implemented to find an optimum route from source to destination. Each route is segmented and called as road segment. We investigate environmental, vehicles and road related information for each road segment and generate a road weight for this particular road segment using Deep-Neuro-Fuzzy model [8][20][21][22]. However, ITMS needs some simulation or emulation to provide that the system works accurately. In this paper, we are proposing a traffic simulation environment by integrating Sumo Urban Mobility Simulation (SUMO) tool and a real-time Intelligent Traffic Management System (ITMS) [8] [11] [20] [21] [22]. Such simulations are expected to offer the prediction and on-the-fly feedback for better decision-making on complex traffic managerial issues and incorporate them to the end-user applications. Simulation or emulation based ITMS implementation can explore an opportunity to expose the model and proof the workability of the model. In our

simulation environment, the full simulation is modeled into a sample map. Sample map can be drawn by SUMO NETEDIT software or modified manually for all roads, vehicle speeds, and path directions. The real map of Gulshan, Dhaka is collected from the internet as “.osm” format, edited it according to the road segments, and then placed it for simulation. A new Graphical User Interface (GUI) is developed to control the simulation. This GUI accesses the value of the attributes and takes input from the user. User can set random values to the decision attributes, or IoT tool collects attributes values directly from the environment and vehicles. Deep-Neuro-Fuzzy framework [8] is used to classify those values and to formulate a weight for a mapped road segment. We expose a multi-agent-based optimum route-finding algorithm- ACO based System into SUMO simulator tool to make the ITMS more robust and self-adaptive. Thus, we need to modify and adjust the ACO according to our ITMS and SUMO implementation. We consider road segments as edges or links and road speeds are trails or pheromone levels. In our implementation, ACO model is analyzed by changing internal parameters including pheromone density, pheromone trail, visibility, and optimum setting for control parameters ( $\alpha$ ,  $\beta$ ) to route more vehicles from source to destination within a certain period. After that, we compare the performance of ACO with existing SUMO based route finding algorithms including CHWrapper, A\*, and Dijkstra. The results highlight that the ACO model performs better than other existing models.

The remainder of the work is organized as follows: Section II introduces the related works and state-of-the-art technologies. Section III gives the implementation details of the proposed simulation model. Section IV shows the result analysis and performance comparison between different routing algorithms. Section V discusses the challenges face to achieve the problems and their solutions. Finally, the conclusion is presented in the final section.

## II. RELATED WORKS

Machine learning techniques are used in [1] [14] [15] to reduce transportation time by optimizing travel options or paths. On the other hand, communication with sharing traffic information is also helped to solve the transportation constraints [2][3] and improve travel quality in both time and speed domains. In this regard, Google developed a traffic condition detection mapping technique [13] using GPS enable

Google Map API and crowd sourcing [16] to collect metadata from user devices anonymously. However, it violates user privacy during Street View Mapping project by scooping personal information, including passwords and e-mail related metadata. Thus, another Traffic Management System (TMS), with optimum route prediction, but exclusive of the privacy violation is required. We proposed an ITMS and implemented it in a piecewise fashion. Besides, the model does not have any empirical evidence of working the whole model. Therefore, the proposed ITMS needs some comprehensive analysis, other means some simulation or emulation, etc., to provide that the hypothesis is valid.

SUMO [4][5] is a simulation framework that is used for microscopic and continuous road traffic simulation package designed to handle large road networks. It applies to develop different applications, including online traffic monitoring system [17], traffic light games [18], and etc. Two different modules Sumo-net-convert and Sumo-route, and they are added to get the best routing decision. NS2[19] is another network simulation tool but usually used for network simulation purpose mainly in node to node connection in different topologies. It also has lacked on the user interface for visual traffic simulation. Thus, SUMO is chosen for implementing our proposed Traffic Management System (TMS). Four (4) optimum route-finding algorithms Dijkstra [9], A\*[9], and CHWrapper [9] are already used as built-in API and integrated with SUMO software.

A significant number of researches are conducted using ACO for path detections and routing decisions. For instance, strategy for optimal truss design [23], solution of TSP (Travelling Salesman Problem) [24], optimization for the asymmetric travelling system [25], and Dynamic Vehicle Routing Problem analysis [26] [27]. Previously, our ITMS system was used to find optimum routes using the Dijkstra algorithm, and there was no real-time simulation to visualize the optimum path. Thus, SUMO based simulation will help to simulate the vehicle routing paths and proves the workability of the model. Besides, SUMO can port the Vehicle Ad-hoc Network with different machine learning algorithms including ACO, fuzzy logic, and etc. and expose a comparative route-finding environment for ITMS. In these regards, this is the first attempt to integrate ACO with SUMO and ITMS, and open an opportunity to compare ITMS performance with existing SUMO built-in route-finding models.

### III. SUMO SIMULATOR SETUP FOR ITMS

Sumo Urban Mobility Simulator (SUMO) provides an opportunity to explore the ITMS in realistic style with a significant focus on traffic related issues including route choice, simulate traffic light, vehicular communication, and etc.

#### A. Implementing Road Maps in Sumo

A certain part of Dhaka city map is downloaded from the internet (openstreetmap.org) as .osm format (Fig- 1). At first, ITMS simulation is implemented in a prototype map (Fig- 2) which is selected from the real map (Fig- 1). After that, the whole simulation is implemented on the real map of Dhaka city. Sumo needs to convert the map files to XML format for

generating optimum route and traffic system. DOS command script “start-command-line.bat” and “netconvert” commands are used to convert the map. osm files to map.net.xml files. Additional necessary elements including road name, traffic light timing, etc. are added using Sumo Neteditor. For vehicle simulation on SUMO needs to configure the configuration file on “map.sumocfg”, and the environmental interface of SUMO can be changed in “gui-setting.cfg” file [10]. Using Sumo GUI software, we simulate both maps and observe the vehicle simulation at the initial stage (Fig- 3 and Fig- 4).



Fig. 1. Map of Gulshan, Dhaka (Real Map).



Fig. 2. Sample Map.



Fig. 3. Initial Simulation Result for Real Map.

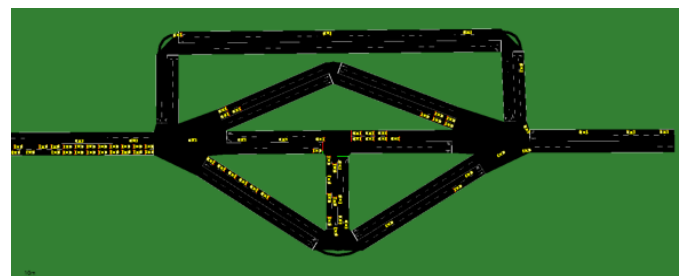


Fig. 4. Initial Simulation Result for Sample Map.

B. Integrating Road Weight Module

A new GUI is developed (Fig- 5) in C# to make the simulation process interactive and user-friendly. This GUI helps to enhance the testing process by changing different weights to the system and thus control the overall traffic flow simulation. Weight calculation depends on eight (8) different attributes including humidity, peak hour, rainfall, temperature, wind, road status, road construction and road accident. All these attributes are combined with varying ranges, as presented in Table I. The selected attributes from GUI are placed to generate the weight value by C-means clustering and Deep-Neuro-Fuzzy software [8]. The weight output range is 0-10. "0" is the lowest weight value that means the natural and road conditions are in the best situation; for that reason, the roads get the maximum speed to ensuring faster vehicle passing. "10" is highest weight value, which means the nature, or road, or both are in an odd situation, for that reasons the roads (or lanes on-road) get the lowest speed value and vehicle passing would be stopped or reduced their speed. Fig- 6 presents the relationship chart between weight and speed. Using Deep-Neuro-Fuzzy tool, we get weight values for each road segment and can be set for specific roads/ a single road/ a single lane. The sample of input data and their corresponding weights and speeds are presented in Table II.



Fig. 5. GUI for Attributes Value Inputting.

TABLE. I. RANGE OF ATTRIBUTES

Attributes	Range	Value Named As
Humidity	32-95	
Peak Hour	8-129	
Rainfall	0-30	
Temperature	13-34	
Wind	0-15	
Road Status	0,1	normal
Road Construction	0,1,2	low, medium, high
Road Accident	0,1	no, yes

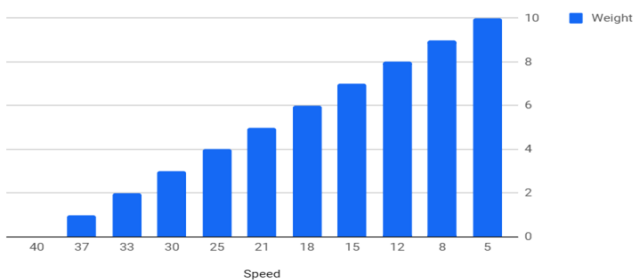


Fig. 6. Weight vs Speed.

TABLE. II. WEIGHT AND SPEED VALUES (FEW SAMPLES)

Attributes								Calculated Weight	Speed Across Weight
Humidity	Peak Hour	Rainfall	Temperature	Wind	Road Status	Road Construction	Road Accident		
80	15	4	21	5	0	0	0	3	30
70	40	0	28	2	0	0	0	7	15
60	78	0	24	1	0	1	0	9	8
65	10	0	20	4	0	0	1	10	5

C. Modeling Ant Colony Optimization (ACO)

ACO was proposed by Macro Dorigo in 1992 [24]. It is used to search for an excellent solution to a given optimization problem. In ACO, a set of software agents called artificial ants or simply ants and they work together to construct a solution by moving on the problem graph but biased by a pheromone model. Ants use the amount of pheromone (a set of parameters is associated with graph edges) to determine the path for gathering their food and caring them back to the nests through an optimum way. Especially they follow that path which has the highest density of pheromone, and after a certain period the amount of pheromone will be vaporized, and then ants may choose another route to reach the destination. Thus, ant pheromone is the key to make ACO adaptive as well as self-learning technique and it can be calculated as:

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{1}{L_{i,j}} k^{th} \text{ ant travel on edge } (i,j) & \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Where i and j are the two (2) nodes of edge (i, j) and k is the particular ant number.  $\Delta\tau_{i,j}^k$  is the amount of pheromone that is created by the k<sup>th</sup> ant in (i, j) edge. L is the length between (i, j).

The total amount of pheromone in a particular edge (i, j) will be the summation of all amount of pheromones are created by all ants (m) travel through link (i, j) and represented as:

$$\tau_{i,j}^k = \sum_{k=1}^m \Delta\tau_{i,j}^k \text{ Without vaporization} \quad (2)$$

However, pheromone can be evaporated during time. Thus, the above equation transforms to:

$$\tau_{i,j}^k = (1 - \rho)\tau_{i,j} + \sum_{k=1}^m \Delta\tau_{i,j}^k \text{ With vaporization} \quad (3)$$

Where  $\rho$  is the evaporation rate, and  $\rho = 0$  means no evaporation and then the amount of pheromone in an edge/link old, and new pheromone  $\rho = 1$  means all old pheromones are evaporated. Thus, the current pheromone will be equal amount to the new pheromone.

$$P_{i,j} = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum ((\tau_{i,j})^\alpha (\eta_{i,j})^\beta)} \quad (4)$$

where  $\eta_{i,j} = \frac{1}{L_{i,j}}$

$P_{i,j}$  is the transition probability from  $i$  to  $j$  for  $k^{\text{th}}$  ant.  $\alpha$  and  $\beta$  are parameters that control the importance pheromone trail (lots of traffic and it is highly desirable) or visibility (close town should be chosen with a higher probability).

Using the above ACO concept, we develop the simulation environment with our prototype map with some modification in road segments. We consider road segments as edges or links and road speeds are trails or pheromone levels. When the road is busy, weight is being higher, speed is lower, and vehicles simultaneously avoid that path and vice versa. Meanwhile the nearby roads weight may be lower, or speed may be higher, and vehicles select those paths to avoid that congestion, and this road may also get busy after some time (due to passing more vehicles to the road segment, weight becomes higher, or speed becomes lower). Then again vehicles select other low weighted, high speed road segments and this procedure will continue. Thus, we are using both pheromone trail and visibility to generate the transition probability  $P_{i,j}$ . Visibility is decided by the value of  $\eta_{i,j}$  and it is the inverse of distance ( $L_{i,j}$ ) between two road segments.

#### D. Implementing ACO in SUMO

This work is the first and foremost attempt to implement ACO in SUMO environment. SUMO is not initially designed for simulating automated vehicles, and we present an interface for exchanging the road weights generated by the Deep-Neuro-Fuzzy software model between ITMS to SUMO environment. Besides, SUMO is not directly computable with algorithms that require dynamic runtime event to change SUMO's internal environmental state. We apply an indirect approach-restart method to hold that dynamicity.

SUMO needs to make a starting call with its local built-in modules to execute any external user-defined modules (e.g., ACO). Thus, we choose A\* algorithm to initialize traffic simulation, and after a certain period, the whole simulation restarts from the last saved state with ACO algorithm. The following code segments are used to initialize SUMO environment and make the starting call to run the A\* algorithm.

```
process_timer.StartInfo.FileName="cmd.exe";  
process_timer.StandardInput.WriteLine("sumo-  
gui --routing-algorithm astar --start -c  
test7.sumocfg");
```

After completing the initial stage, the simulation system needs to restart and generates several .xml files during simulation where each record contains the state of the simulation process. Thus, the last file contains the most updated state of the simulation and is required for successive simulations.

SUMO needs to be stopped before restarting the next phase execution with loading the previous phase's last saved state (last output file) information. Now the question is –what will be the suitable time duration of an execution phase of the simulation model? We experimented and traced that SUMO takes 2/3 seconds for each restart. Thus, 10 seconds duration in each execution phase cannot demonstrate the simulation impact properly. However, 30/40 seconds duration takes

longer time to demonstrate. 20 seconds seems enough to balance the above mentioned constrains. In addition, this time duration also add constrains to data collection and processing interval to 20 seconds.

Since the system needs to take multiple times of restarting for applying ACO (because the speed of road segment may need to be changed after a certain period of time), a timer function is implemented, the interval time is 20 seconds for restarting:

```
processTimer.Tick+=newEventHandler(processTime  
r_Tick);  
processTimer.Interval=20*1000;
```

Usually, ACO directs the vehicle to the best optimum route, and chooses the same direction, until the timer stops the simulation. Our ACO implementation can handle uncertain situations including heavy rainfall, peak hour, road construction, or accident. During the precarious conditions, the weight of the road segment will be changed and also changed the road speed. Then, other road segments will get the best road speed and continue the vehicle flow. The situation is adequately explained in Fig- 7 and 9. Fig- 7 shows the normal position, a smooth vehicle flow in the system. After a little moment, a road accident is occurred (created by GUI in Fig- 8) in the road segment named edge2.

Our simulation model will detect the occurrence road accident and determine the road segment attribute values accordingly and input them to the Deep-Neuro-Fuzzy model to generate a corresponding weight value of the road segment. Thus, the road segment weight will be adjusted according to accident and reduce the vehicle flow on that particular road segment. Vehicle flow will transfer to the next available route/road segment chosen by the ACO algorithm (in Fig- 9). It will almost reduce all vehicle congestion in the road network as well.

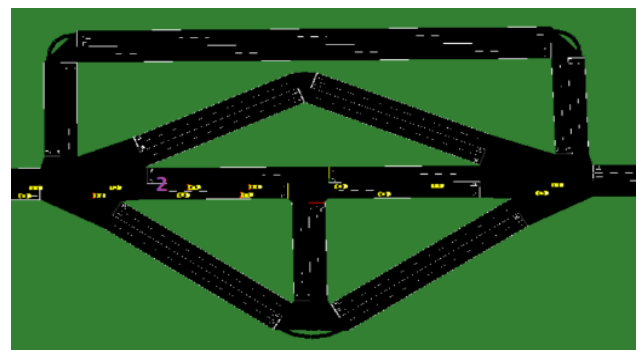


Fig. 7. Smooth Vehicle Flow.



Fig. 8. Selecting Road Accident in Edge2.

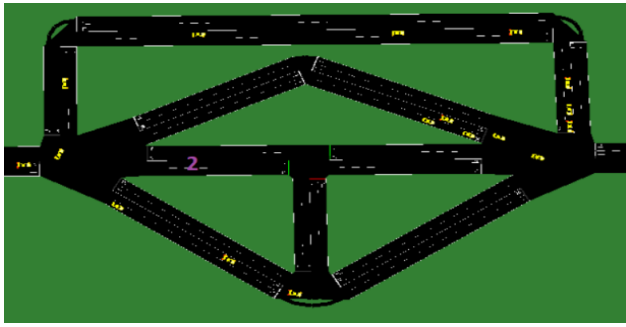


Fig. 9. ACO Choses Alternate Paths and Transfer Vehicle Flow.

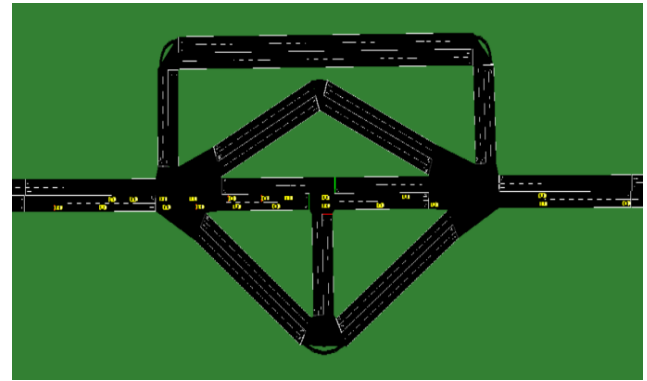


Fig. 10. ACO is Crossing Same Road Segment with Fixed  $\alpha$  and  $\beta$  Values.

#### IV. RESULT ANALYSIS

##### A. Experiments on ACO Parameters

As we mentioned earlier,  $P_{i,j}$  is the transition probability from  $i$  to  $j$  for  $k^{\text{th}}$  ant/car, road speed reflects to trails or pheromone levels and directly proportional to  $P_{i,j}$ . Thus, intuitively high road speed (low weight) segment will be chosen for routing. However, cars will follow the high road speed segments and increase the pheromone level according to ACO algorithm. Certainly, after a period of times the road segment will be congested. Then our ITMS automatically change the road segment weight to higher means low pheromone level will be in that road segment. Thus, we do not need to create any vaporization effect in our ACO implementation. Nonetheless, the road segment weight will not be changed during the 20 seconds simulation periods. Because, the restart method needs 20 seconds duration to update the next road weights and creates effect in the simulation. Thus,  $\alpha$  and  $\beta$  parameters values will play an important role within the 20 seconds simulation periods to produce the real effect of ACO.  $\alpha$  and  $\beta$  control the road segment probability according to pheromone trail or visibility. Choosing higher  $\alpha$  value means choosing denser pheromone trail and it reflects to lot of traffics. Other side, choosing higher  $\beta$  means choosing closer road segments. There are two (2) concerning issues in this regard.

1) fixed  $\alpha$  and  $\beta$  will choose a road segment with highest probability and it will continuously be chosen according to the ACO algorithm except the vaporization occurs or weight/speed is going to be readjusted due to the external effects. After a certain period of times the traffic on this road segment will be increased and congestion will occur. We experimented in SUMO with  $\alpha=0.1$  and  $\beta=0.1$  fixed value to simulate 200 vehicles with ACO algorithm. Thus,  $P_{i,j}$  will be increased in a particular road segment and vehicle flow will also be increased. Due to the fixed value in  $\alpha$  and  $\beta$  path/road segment will not be changed. The situation is clearly stated in Fig- 10. Where the vehicles are choosing same road segment continuously and congestion situation may arrive soon. To avoid such circumstances, we need to decrease the vehicle speed automatically after a certain period of times. Thus, we need to reduce  $\alpha$  value so that alternate path can be selected to avoid such situation in our ITMS.

2) Dynamic changing in  $\alpha$  and  $\beta$  values during execution of ACO is required. We also need to find out the changing techniques/steps of  $\alpha$  and  $\beta$ , and trace their optimum conditions.

Thus, we do grid search on  $\alpha$  and  $\beta$  values in SUMO and trace the optimum situation for ACO algorithm to route 200 vehicles in a shorter period of times. Table III presents the grid search results, and according to the result setting  $\beta=1-\alpha$  value almost find the optimum time. To avoid starvation or congestion in our ACO implementation, in each step, we decrease  $\alpha$  value to 0.05 to the highest  $P_{i,j}$  road segment, and increase  $\alpha$  value to 0.05 to other nearby road segments. The above situation is implemented with our proposed ACO, and the congestion situation is by passed and simulated more vehicles during a fixed period of times. Fig- 11 presents snapshot of the updated simulation.

##### B. Performance Evaluation of ACO with Built-in Route-Finding Algorithms (Dijkstra, A\*, CHWrapper)

We develop an experimental setting with our SUMO based ACO implementation and other three (3) built-in SUMO route finding algorithms (Dijkstra, A\*, CHWrapper) on our sample maps. The objective of this experimental setup is to evaluate the performance of all algorithms to simulate/schedule 500 vehicles from source to destination (one way only). Table IV presents the performance of those algorithms in time unit (seconds). The time unit can be represented as second, Nano seconds, mile seconds etc. Depending on the computer specification, the simulation time unit will be varied. Among the built-in algorithms in Sumo environment Dijkstra performs poorer than others. However, it is a simple algorithm and easy to implement. That is why it has been chosen as the default path finding algorithm in SUMO environment. Contraction Hierarchies Wrapper (CHWrapper) is a preprocessing-based routing algorithm. As expected, it performs better than other two (2) A\* and Dijkstra implementations. However, comparing ACO with CHWrapper, ACO performs better than CHWrapper in most of the test cases. ACO takes 1580 unit of time on average of 100 simulations with  $\alpha$  values between 0.1-0.9 and  $\beta=1-\alpha$  values. Overall, we can conclude that ACO performance is better than all other algorithms for vehicle routing.

TABLE. III. GRID SEARCH ON  $\alpha$  AND  $\beta$  VALUES

$\alpha$	$\beta$	Time (Second)	$\alpha$	$\beta$	Time (Second)	$\alpha$	$\beta$	Time (Second)
0.1	0.1	707	0.4	0.1	721	0.7	0.1	730
0.1	0.2	702	0.4	0.2	711	0.7	0.2	735
0.1	0.3	698	0.4	0.3	713	0.7	0.3	727
0.1	0.4	699	0.4	0.4	716	0.7	0.4	739
0.1	0.5	686	0.4	0.5	705	0.7	0.5	732
0.1	0.6	653	0.4	0.6	701	0.7	0.6	728
0.1	0.7	634	0.4	0.7	699	0.7	0.7	740
0.1	0.8	621	0.4	0.8	708	0.7	0.8	736
0.1	0.9	616	0.4	0.9	704	0.7	0.9	734
0.2	0.1	714	0.5	0.1	722	0.8	0.1	720
0.2	0.2	713	0.5	0.2	731	0.8	0.2	714
0.2	0.3	699	0.5	0.3	729	0.8	0.3	703
0.2	0.4	687	0.5	0.4	725	0.8	0.4	701
0.2	0.5	650	0.5	0.5	730	0.8	0.5	710
0.2	0.6	644	0.5	0.6	719	0.8	0.6	722
0.2	0.7	640	0.5	0.7	713	0.8	0.7	738
0.2	0.8	642	0.5	0.8	720	0.8	0.8	753
0.2	0.9	651	0.5	0.9	707	0.8	0.9	744
0.3	0.1	715	0.6	0.1	720	0.9	0.1	730
0.3	0.2	104	0.6	0.2	725	0.9	0.2	714
0.3	0.3	721	0.6	0.3	718	0.9	0.3	722
0.3	0.4	718	0.6	0.4	716	0.9	0.4	734
0.3	0.5	692	0.6	0.5	727	0.9	0.5	728
0.3	0.6	701	0.6	0.6	735	0.9	0.6	731
0.3	0.7	695	0.6	0.7	731	0.9	0.7	737
0.3	0.8	710	0.6	0.8	728	0.9	0.8	742
0.3	0.9	705	0.6	0.9	730	0.9	0.9	749

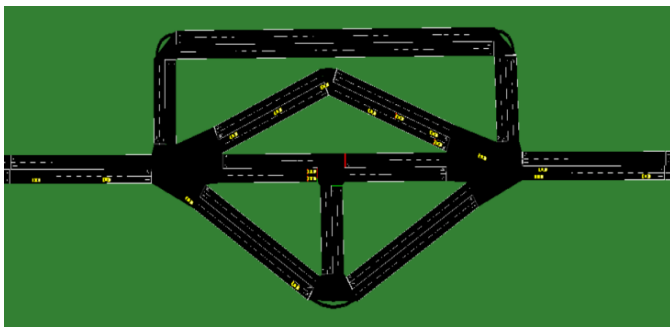


Fig. 11. ACO with Dynamic  $\alpha$  and  $\beta$  Values.

TABLE. IV. PERFORMANCE COMPARISON OF DIJKSTRA, A\*, CHWRAPPER AND ACO ALGORITHMS

Algorithms	Time (Seconds) For 500 vehicles
Dijkstra	1840
A*	1675
CHWrapper	1630
Ant Colony	1580

## V. DISCUSSION

A novel Deep-Neuro-Fuzzy classification model [7] [8] was implemented in python language with TensorFlow library support to solve the weight overlapping problems, to remove the data outliers, and to avoid the overfitting problems and 98.63% classification accuracy. We used the Deep-Neuro-Fuzzy model to calculate dynamic road weights using environmental, road and vehicle-related decision attributes. However, the model did not have any empirical evidence of working in the traffic system. Simulation or emulation-based implementation can explore an opportunity to expose the model and proof the hypothesis. On these regards, SUMO open-source tool was used to simulate the total ITMS.

In addition, a new model- Ant Colony Optimization (ACO) was implemented with SUMO tool to support multi-agent based collaborative decision-making environment for ITMS. We also compared the performance of the proposed ACO model with other built-in simulation models (CHWrapper, A\*, and Dijkstra) in SUMO simulation environment and tested the model work abilities. Our results highlighted ACO for its better performance than others.

Although SUMO is an open source simulation framework, however, it has a complex design and architecture, and does not support any upgradation or modification into its internal built-in route findings modules. Thus, our first and foremost challenge is to implement a new algorithm module (ACO model) into the SUMO environment. We solved the problem using C# base scripts to execute the ACO module explicitly then port our algorithm into SUMO.

The second challenge we faced to incorporate the ACO into our ITMS system, especially with Deep-Neuro-Fuzzy model. We solved the problem by considering road segments as edges and road speed or road weight as pheromone level. Thus, we simulated the ITMS in pragmatic style with a micro traffic, open source traffic simulation model (SUMO), and solved the challenges related to modeling and simulation for ITMS. Article in [6] also present a similar approach to solve the traffic management problem. However, they used fuzzy logic and ACO on one attribute (# of vehicles on a road segment) only to detect congestion amount of a particular road segment. They did not focus other relevant attributes including weather, road or vehicle. In addition, counting # of vehicles in a segment is easy to find in simulation environment but difficult in real scenarios. Because the process requires instantaneous congestion detection method, which is not discussed in [6]. We also did a comparative study with other methods and implemented a low-cost but flexible ITMS method in [20][21][22]. Our Deep-Neuro-Fuzzy model [7][8] uses multivariate and multi-attributes data and simulates properly in SUMO environment. Our IoT based embedded sensor system helps the ITMS to collect the real-time environment, road and vehicle related information with low cost, less bandwidth, but flexible environment. However, the complete ITMS or even the Deep-Neuro-Fuzzy model is not yet been implemented in a real test bed with real traffic scenarios and their workability is still unproved in real road networks.

In addition, a central server approach (for data uploading, processing, and sending results back to the server) was proposed in [6] and all our previous works [11][12][20][21][22][30]. However, central server approach will create great pressure on the network for large number of sensors or IoT devices and massive data processing situations. Thus, edge and fog computing [28] [29] will provide a novel approach in this research domain and support a load balancing, power balancing, and mobile communication network for our ITMS.

## VI. CONCLUSION

ITMS is successfully implemented in SUMO with traffic related issues including route choice, traffic light simulation or vehicular communication and etc. The road weight generate module- Deep-Neuro-Fuzzy tool was suffering for an interface to provide input data to the ITMS. SUMO based ITMS supports a GUI for Deep-Neuro-Fuzzy tool and is capable to change the weight according to road and vehicle related attributes and helps to improve route situation.

ACO is successfully implemented as an external SUMO module and compare its performance with other built-in route-finding modules. Three (3) different parameters including road

speed,  $\alpha$  and  $\beta$  are explored and investigated in vehicle routing. ACO takes 1580 unit of time on average of 100 simulations with  $\alpha$  values between 0.1-0.9 and  $\beta=1-\alpha$  values. ACO based ITMS performs better and suggest for future use.

## VII. FUTURE WORKS

In future, route finding algorithms can be explored with reinforce learning (Q-learning) algorithm and evaluated performance in our ITMS. IoT based input devices will be connected directly to take different real time weather and vehicle related data from different road segments. In near future we will try to implement the full ITMS in Dhaka City, Bangladesh. VANET (Vehicular Ad-Hoc Network) can also be another option to incorporate in our proposed system where vehicles will communicate among themselves to share road status, road accident using wireless connection. Integrating ACO algorithm and VANET into vehicles and manage them through our ITMS is the future goal of our research.

## ACKNOWLEDGMENT

This research is supported and funded by the Information and Communication Technology (ICT) Division, Government of the People's Republic of Bangladesh. The authors would also like to show their gratitude International University of Business Agriculture and Technology (IUBAT), Sector-10, Uttara Model Town, Dhaka, Bangladesh for their necessary supports.

## REFERENCES

- [1] P. Yousefi and R. Zamani, "The optimal routing of cars in the car navigation system by taking the combination of divide w Journal of Machine Learning and Computing (IJMLC), Vol. 3, Issue 1, February 2013, pp. 44-48..
- [2] S. Inoue, Y. Taoda and Y. Kakuda, "An Alleviating Traffic Congestion Scheme Based on VANET with a Function to Dynamically Change Size of Area for Traffic Information in Urban Transportations," 10th International Symposium on Autonomous Decentralized Systems (ISADS), 2011, pp. 299-302.
- [3] M. Kimura, S. Inoue, Y. Taoda, T. Dohi and Y. Kakuda, "A Novel Method Based on VANET for Alleviating Traffic Congestion in Urban Transportations," IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 2013, pp. 1-7.
- [4] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO- Simulation of Urban Mobility", International Journal on Advances in Systems and Measurements, vol 5 no 3 & 4, year 2012, pp.128-138.
- [5] Krajzewicz, Daniel & Hertkorn, Georg & Feld, Christian & Wagner, Peter. "SUMO (Simulation of Urban MObility); An open-source traffic simulation". 4th Middle East Symposium on Simulation and Modelling (MESM2002), 2002, pp.183-187.
- [6] F. Khodadadi, S. J. Mirabedini, A. Harounabadi, "Improve Traffic Management in the Vehicular Ad Hoc Networks by Combining Ant Colony Algorithm and Fuzzy System", International Journal of Advanced Computer Science and Applications(IJACSA), vol. 7, No. 4, 2016, pp. 44-53.
- [7] S. Akhter, S. Sumit, M. R. Rahman "Design and Implementation of an Intelligent Traffic Management System- A Neural Approach", In book: Exploring Critical Approaches of Evolutionary Computation, Chapter: 4, pages 58-79. Publisher: IGI Global, 10.4018/978-1-5225-5832-3.ch004, 2018, pp.58-79.
- [8] S.H. Sumit and S.Akhter, "C-Means Clustering and Deep-Neuro-Fuzzy Classification for road weight measurement in traffic management system", Soft Computing, Springer, Vol 23, Issue 12, June-2019, pp 4329-4340. <https://doi.org/10.1007/s00500-018-3086-0>.

- [9] V. T. Nha , S. Djahel, & J. Murphy. "A comparative study of vehicles routing algorithms for route planning in smart cities". 2012 First International Workshop on Vehicular Traffic Management for Smart Cities (VTM) 2002.
- [10] Çelikoğlu, Hilmi & Dobrowiecki, Tadeusz & Alms, Robert & Klischat, Moritz & Krol, Jakub & Chraïbi, Mohcine & Saïvichit, Chaiyachet & Banijamali, Ahmad & Hafner, Alexander & Behrisch, Michael & Bieker-Walz, Laura & Bonne, Jean-Benoît & Vullings, Erik & Schott, Benedikt & Härrï, Jérôme & Weber, Thomas & Erdmann, Jakob & Flötteröd, Yun-Pang & Grigoropoulos, Georgios & Wesemeyer, Daniel. (2019). SUMO User Conference 2019.
- [11] S.Akhter, M. R, Rahman & A. Islam. "Neural Network (NN) Based Route Weight Computation for Bi-Directional Traffic Management System". International Journal of Applied Evolutionary Computation. Vol. 7. issue 4. 2016. pp.45-59.
- [12] M. A. A. Forhad, M. Nadim, M. R. Rahman and S. Akhter , "Cloud IoT based Mobile Agent Framework for real-time traffic information acquisition, storage and retrieval, "Smart Devices, Applications, and Protocols for the IoT", Chapter-II, 2019 PP 14-32, IGI Global publisher.
- [13] B.W. Dekock, K. L.Russell, R. J.Qian.US6466862B1-Method and apparatus for automatic traffic conditions data collection using a distributed automotive computing system:Oct.15,2002.
- [14] P. Chhatpar, N. Doolani, S. Shahani, & R. Priya. "Machine Learning Solutions to Vehicular Traffic Congestion". 2018 International Conference on Smart City and Emerging Technology (ICSCET). doi:10.1109/icscet.2018.8537260.
- [15] A. Elfar, A. Talebpour, and H.S. Mahmassani, "Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment", Transportation Research Record: Journal of the Transportation Research Board, vol 2672, Issue 45, 2018, pp. 185-195. doi:10.1177/0361198118795010.
- [16] L. Estes, D. McRitchie, J. Choi, S. R. Debats, T. Evans, W. Guthe, K. Caylor, "A platform for crowdsourcing the creation of representative, accurate landcover maps", Vol 80, June 2015, pp. 41-53.
- [17] M. Krumnow, A. Kretschmer, "Real-Time Simulations Based on Live Detector Data – Experiences of Using SUMO in a Traffic Management System", Simulation of Urban Mobility Lecture Notes in Computer Science,vol. 8594, 2013, pp. 136-145. doi:10.1007/978-3-662-45079-6\_10.
- [18] Y. Xu, D. Li, Y. Xi, S. Lin,"Game-Based Traffic Signal Control with Adaptive Routing via V2P", 21st International Conference on Intelligent Transportation Systems (ITSC), 2018. doi:10.1109/itsc.2018.8569587.
- [19] A. Gupta, H. Bedi, M. D. B., V. Shashidhar (2010). Simulation to track 3D location in GSM through NS2 and real life. International Journal on Applications of Graph Theory In Wireless Ad Hoc Networks And Sensor Networks,vol 2 issue1, 2010, pp. 18-29.
- [20] M.R. Rahman and S. Akhter, "Real time bi-directional traffic management support system with GPS and websocket", In: Proc. of the 15th IEEE International Conference on Computer and Information Technology (CIT2015), Liverpool. UK,2015.
- [21] M.R. Rahman and S. Akhter, "Bidirectional traffic management with multiple data feeds for dynamic route computation and prediction system", International Journal of Intelligent Computing Research (IJICR), Special Issue. Vol. 7, Issue 2, 2015, PP. 720-727.
- [22] M. R. Rahman and S. Akhter, "Bi-directional traffic management support system with decision tree based dynamic routing", In: Proc. of 10th International Conference for Internet Technology and Secured Transactions, ICITST, 2015, London. United Kingdom.
- [23] S. Christodoulou, "Optimal truss design using ant colony optimization", In: Proc. of 5th GRACM International Congress on Computational Mechanics, Limassol, 2005.
- [24] M. Dorigo, L. Gambardella,"Ant colony system: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, Vol. 1, Issue 1, 1997, PP. 53-66. doi:10.1109/4235.585892.
- [25] M. Dorigo, V. Maniezzo and A. Coloni, "Ant system: Optimization by a colony of cooperating agents", IEEE Trans. Syst., Man, and Cybern, Part B. Vol. 26, 2002, PP. 29-41.
- [26] X. Haitao, P. Pan and D. Feng., "Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization". Discrete Dynamics in Nature and Society, 2018, PP. 1-13. 10.1155/2018/1295485.
- [27] D. Coltorti, and A.E. Rizzoli, "Ant colony optimization for real-world vehicle routing problems". ACM SIGEVolution, vol. 2 issue. 2, 2017,pp. 2-9. doi:10.1145/1329465.1329466.
- [28] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin and X. Yang, "A survey on the Edge Computing for the Internet of Things", IEEE Access, vol. 6, Mar 9, 2018, pp. 6900-6919. doi: 10.1109/ACCESS.2017.2778504.
- [29] Y. Jararweh, A. Doulat, O. AlQudah, E. A., M. Al-Ayyoub, and E. Benkhelifa, "The Future of Mobile Cloud Computing: Integrating Cloudlets and Mobile Edge Computing", 23rd International Conference on Telecommunications, 16-18 May, Greece, 2016.
- [30] S. Nawrin, M. R. Rahman and S. Akhter, "Exploreing k-means with internal validity indexes for data clustering in traffic management system" Thomson Reuters Master Journal List and ISI index. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 8, No. 3, March 2017, pp. 264-272.