



# Preemptive MACO (MACO-P) Algorithm for Reducing Travel Time in VANETs

Vinita Jindal & Punam Bedi

**To cite this article:** Vinita Jindal & Punam Bedi (2017) Preemptive MACO (MACO-P) Algorithm for Reducing Travel Time in VANETs, Applied Artificial Intelligence, 31:2, 174-196, DOI: [10.1080/08839514.2017.1300017](https://doi.org/10.1080/08839514.2017.1300017)

**To link to this article:** <https://doi.org/10.1080/08839514.2017.1300017>



Published online: 31 Mar 2017.



Submit your article to this journal [↗](#)



Article views: 242



View related articles [↗](#)





View Crossmark data [↗](#)



Citing articles: 3 View citing articles [↗](#)

## Preemptive MACO (MACO-P) Algorithm for Reducing Travel Time in VANETs

Vinita Jindal  and Punam Bedi 

Department of Computer Science, University of Delhi, New Delhi, India

### ABSTRACT

Vehicular Ad-hoc NETWORKS (VANETs) are extremely flexible and dynamic ad-hoc networks that are used to provide smooth, safe and comfortable journey to commuters. The commuters spend significant time of their journey either for their turn to cross the intersections or in the congestion on the path. For reducing the total travel time, it is essential to minimize the waiting time at intersections and find best/optimal congestion-free paths for smoother movement of vehicular traffic on roads. A novel Preemptive Modified Ant Colony Optimization (MACO-P) algorithm has been proposed in this paper for reducing the total travel time. The Modified Ant Colony Optimization (MACO) algorithm is used in literature to avoid the congested path by sensing the pheromone trail. Adding preemption to the existing MACO algorithm will result in the reduction of the average queue length at intersections, meaning thereby, less waiting time ensuring smooth mobility of vehicles. For implementation, various open source softwares, i.e., OSM, Simulation of Urban MObility (SUMO), MObility model generator for VEhicular networks (MOVE), Python and Traffic Control Interface (TraCI) are being used. Real-time maps are being fetched by OSM. Traffic simulation is done using SUMO and the Mobility model is generated through MOVE. Python is used for writing client scripts that initiate and control the simulation, with traffic control interface provided by TraCI. Experimental results confirm that the total travel time is reduced using the proposed MACO-P algorithm, resulting in significant reduction in consumption of fuel.

### Introduction

An ad-hoc network with moving vehicles as nodes and distance between these vehicles on roads as edges is known as Vehicular Ad-Hoc Network (VANET). Each vehicle in VANET can exchange information across the wireless medium with other vehicles or with the existing fixed infrastructure (Altayeb and Mahgoub 2013). Vehicle to Infrastructure (V2I), Infrastructure to Infrastructure (I2I) and Vehicle to Vehicle (V2V) are three types of

**CONTACT** Vinita Jindal  [vjindal@keshav.du.ac.in](mailto:vjindal@keshav.du.ac.in)  Department of Computer Science, University of Delhi, Delhi 110007, India.

Color versions of one or more of the figures in the article can be found online at [www.tandfonline.com/UAAI](http://www.tandfonline.com/UAAI).

© 2017 Taylor & Francis

communications used in VANETs (Liang et al. 2014). Vehicles may follow various mobility patterns depending on the underlying structuring of roads and traffic lights, traffic conditions and driving behaviors of the person on steering wheel (Dhankhar and Agrawal 2014). The vehicles are equipped with Global Positioning System (GPS) devices or sensors that support in supplying information for routing decisions, traffic estimation, signal timings etc. (Rehman et al. 2013). VANET has a remarkable ability to improve fuel efficiency, road safety and comfort for both drivers and commuters at the same time (Kakarla et al. 2011).

VANET is an extremely scalable network with varying number of nodes (vehicles) and road-side units (infrastructure) that keep on changing in real time (Moustafa and Zhang 2009). Network topology also changes very frequently as nodes move with variable speed. VANETs are integrated in intelligent transportation systems to create smart transportation systems (Transportation 2014). Another area for research in VANETs is to reduce the time spent on roads, including travel time as well as waiting time at intersections, for a commuter. Travel time can be reduced by using the shortest path between any two nodes, but if all vehicles select the shortest route, then it will create congestion on that route. Thus, the optimal route is not always the shortest one (Bell and McMullen 2004). Commuters may choose a slightly longer route but their prime concern is to avoid congestion en route by encountering less number of traffic signals and less waiting time for smooth mobility.

This paper proposes a preemptive Modified Ant Colony Optimization (MACO-P) algorithm that optimizes the routes taken by the vehicles through traffic signals and simultaneously reduces congestion on the chosen path, ensuring overall less fuel consumption. This can be achieved by lowering the waiting time at signals and allowing vehicles to choose the optimal path irrespective of the distance covered. Preemptive adaptive traffic signals are used in the presented work to reduce waiting time at intersections and Modified Ant Colony Optimization (MACO) is used for avoiding congestion on roads. The proposed algorithm works under the assumption that all the roads and vehicles are in working condition.

The MACO algorithm (Jindal et al. 2015) modifies the behavior of Ant Colony Optimization (ACO) by selecting the path with minimum pheromone instead of maximum to avoid congestion on the roads. Preemptive adaptive algorithm (Bedi et al. 2015) is used for reducing the waiting time at traffic signals by reducing average queue length at intersections. In the proposed MACO-P algorithm, MACO algorithm is modified and made preemptive, so that waiting time is also reduced along with avoiding congestion. Using the proposed MACO-P, the path chosen by the vehicles may be a bit lengthy but the overall travel time is reduced as waiting time is minimized and congestion is avoided. This is shown experimentally and statistically in

this paper. For implementation, various open source softwares like OSM, Simulation of Urban MObility (SUMO), MObility model generator for VEhicular networks (MOVE), Python and Traffic Control Interface (TraCI) are being used. Real-time maps are being fetched by OSM. Traffic simulation is done using SUMO and the Mobility model is generated through MOVE. Python is used for writing client scripts that initiate and control the simulation with traffic control interface provided by TraCI.

Rest of the paper is organized as follows. “Literature review” section reviews the literature in VANETs related to traffic signals and routing algorithms. The proposed MACO-P algorithm is discussed in “Proposed preemptive MACO (MACO-P) algorithm for reducing travel time” section. The experimental setup and results are presented in “Experimental setup and results” section, followed by statistical inferencing for the proposed MACO-P approach. Finally, “**Conclusion**” section concludes the paper.

## Literature review

Travel time in VANETs can be reduced by lowering either the waiting time at intersections by optimizing traffic signals or the congestion en route by providing the vehicles a smooth and hassle-free route. Traffic signals are used for managing traffic at intersections. These signals can be classified into fixed (or pre-timed) signals and adaptive (or actuated) signals. Fixed traffic signals are operated for pre-decided constant timings (Roess, Prassas, and McShane 2011). Fixed signals can generate the congestion problem, as one direction may have more vehicular traffic in comparison to other directions. With new emerging equipment like detectors and sensors installed on the roads, fluctuating traffic demand can be determined and the signal timings can be adjusted accordingly (Maslekar et al. 2011). These signals are also known as adaptive signals. Dynamic strategies that adjust the signal durations in order to cater to the demand of fluctuating traffic at intersections have been presented in the literature by many authors (Dias et al. 2014; Vasirani and Ossowski 2011). Research on traffic signals is continuously improving and experiencing many alterations due to significant increase in road congestion nowadays (Shivani and Singh 2013; Falocchio and Levinson 2015).

For improving traffic conditions, various guidelines, technologies and new researches are coming up with solutions. To the best of our knowledge, most of the signals use actuated (or adaptive) methodology in a non-preemptive or cyclic manner. The purpose of adaptive traffic signals is to provide hassle-free movement of traffic with less waiting at intersections. Some of the existing adapted traffic light systems are Optimization Policies for Adaptive Control (OPAC) (Gartner 1982), Sydney Coordinated Adaptive Traffic System (SCATS) (Lowrie 1982), Split Cycle and Offset Optimization Technique (SCOOT) (Mimbela and Klein 2000), Real-time

Hierarchical Optimized Distributed Efficient System (RHODES) (Mirchandani and Head 2001) and Real-Time Traffic Adaptive Control Logic (RTACL) (Turner-Fairbank Highway Research Center 2005) to name a few. These systems are implemented through the non-preemptive or cyclic approach for adaptive signals that leads to the problem of unused green time at the phase with no vehicle. It also increases the waiting time for the green time assignment at other phases.

Gradinescu et al. (2007) have proposed another adaptive traffic system that adjusts the signal timing in cyclic order using the current vehicle density on the road. Another such system proposed by Maslekar et al. (2011) also uses the current vehicle density for deciding the traffic signal timings in cyclic order. Khekare and Sakhare (2012) have done a comprehensive survey which illustrates the use of the traffic information obtained through vehicles for avoiding traffic accidents and green time allotment for signals. Pandit et al. (2013) present another traffic control system which is also adaptive and uses real-time information for green time optimization. An adaptive algorithm is also given by authors to reduce the waiting time at intersections by optimizing the green time assignment in their work (Bedi, Jindal and Dhankani, et al. 2015). Most of the adaptive approaches existing in the literature to the best of our knowledge are cyclic. There is a case in which no vehicles are present at the phase with green time allotment. This leads to unused green time for the vehicles waiting on other phases. In such situations, the preemptive approach can be used for providing out-of-order green light assignment to the phase with more traffic and help in reducing the congestion in real time.

An adaptive preemptive and non-cyclic algorithm for traffic signals that is able to minimize the queue length at intersections and hence results in the reduction of waiting time is presented in Bedi et al. (2015). The algorithm calculates the current vehicle density on the road and accordingly assigns the green time to the phase with maximum vehicles or to the phase with vehicles waiting for long. Green light assignment is done by preempting it to cater to the demand as and when required by the phases irrespective of their normal turn. But, after using the preemptive algorithm, vehicles cannot determine their turn for the green time assignment as the green time would not be allotted in the normal round-robin order. If the threshold value is more than the number of vehicles then, vehicles need to spend more time waiting for the green signal. By using this approach, most of the vehicles will get the advantage of getting green time assignment quickly but few may suffer due to small queue at their side. They have implemented the algorithm on simple road network, complex road network and real-time University of Delhi (DU) network. In the present work, we extend the simulation on the network obtained through the real-time map of North West Delhi (NWD) with an increment step of 300 vehicles. The work (Bedi et al. 2015) considered only minimization of waiting time at intersections, which works under normal

road conditions. To handle the case when there is some problem on a specific path, which cannot be handled by this work, we have combined the pre-emptive algorithm with the MACO algorithm for the selection of non-congested routes with less waiting at intersections.

Routing forms an integral part of transportation systems as it helps commuters to reach the destination faster and in the best possible way. It is a very challenging task due to the variable speed and dynamic mobility patterns of the vehicles. Various routing algorithms for VANETs have been proposed in literature by various authors in their work (Lee, Lee, and Gerla 2010; Chaqfeh, Lakas, and Jawhar 2014). Dynamic routing finds an optimal route for vehicles by taking into account the information gained from sensors or other technological equipment (Rana, Rana, and Purohit 2014). Routing algorithms can be categorized based mainly on two parameters, position and topology. Proactive, reactive and hybrid are the three approaches used by the position-based routing algorithms (Li and Wang 2007; Nanda and Das 2011). Dijkstra's algorithm is one of such renowned position-based routing algorithms that finds the shortest path between any two nodes in a network (Altayeb and Mahgoub 2013; Al-Sultan et al. 2014). The algorithm works on local optimum such that it does not target the destination from the start, rather it tries to find the next best possible node from the current node and marks them visited to avoid the cycles. Topology based algorithms are also classified as reactive, proactive and hybrid (Dua, Kumar, and Bawa 2014). Greedy Best First algorithm is a heuristic search technique applied for routing in VANETs (Piran, Rama Murthy, and Praveen Babu 2011). Heuristic is used to produce a solution in reasonable time by using estimation for the underlying problem. This solution may not be the optimal solution. But it is still valuable as it helps to reach the goal quickly (Syal and Kaur 2014). In contrast to Dijkstra's algorithm the next closest node from the current node is not selected, however in every move the next closest node is selected. This algorithm always returns a route that is best because every move is focused on the nodes leading to the goal rather than considering all the nodes that are in the neighborhood of current node.

Swarm intelligence techniques that replicate the behavior of existing living organisms are also used in literature for routing in VANETs (Dempsey and Schuster 2005). Swarms have the unique talent to answer any given real-time and complex tasks which are a challenging to answer through normal algorithms and hence are suitable for the real-time traffic scenario. Ant colony optimization (ACO) is an artificial swarm algorithm that can be used for building paths in communication and networking domains. It has the ability to develop self-organizing methods that help in solving real-time routing problems (Dorigo, Caro, and Gambardella 1999; Ho and Ewe 2009; Claes and Holvoet 2011; Geetha, Vanathi, and Poonthalir 2012). In ACO, a pheromone is left by an ant on its route while passing through that route, which is sensed

by other ants to help them follow the path (Cabanés et al. 2014). Other ants in the group can discover the accumulation of the pheromone and if encountered with higher concentrations of pheromone then, follow that path. Pheromone starts evaporating in the situation where the route remains unused by the vehicles for some time (Dorigo and Thomas 2004). Some of the initial problems that use the ant colonies are quadratic assignments and traveling salesman problem (Elloumi et al. 2014). Several domains exist in the literature that use the ACO technique for various kinds of optimization (Bell and McMullen 2004). One of the approaches discussed for dynamic adaptation of traffic signals using graphs is presented in Bedi et al. (2007). A modified version of ACO for the vehicular traffic environment to yield more optimized results on complex road network is presented in Bin, Zhen, and Baozhen (2009). The core objective is to minimize the waiting time during journey, which in turn reduces the total travel time irrespective of the path length used by the vehicle (Dias et al. 2014).

For reducing congestion on roads, a MACO is proposed by the authors in their paper (Jindal et al. 2015). MACO modifies the pheromone behavior and deviates the vehicles from their chosen path to avoid congestion. Experiments were performed on simple road network, complex road network and real-time DU network to check the effectiveness of the MACO algorithm. The authors in their work have not taken into account, the enroute traffic signals which also increased the overall travel time due to waiting at intersections. In the current work, we have combined the preemptive algorithm with the MACO algorithm, which results in the reduction of the overall travel time during congestion. We compare the proposed algorithm with both existing MACO and the standard Dijkstra's algorithm. The obtained results show a significant reduction in the waiting time after using the proposed MACO-P algorithm, which further decreases the total travel time in VANETs. The proposed algorithm is discussed in the next section.

### **Proposed preemptive MACO (MACO-P) algorithm for reducing travel time**

During a journey, a commuter spends his major time waiting due to either traffic signals at the intersections or congestion en route caused by accidents, some digging work etc. Thus, for reducing the overall travel time, one must reduce the waiting time in both of these areas simultaneously. This paper presents a MACO-P algorithm in VANETs for reducing travel time during road trips. The proposed algorithm reduces waiting time at the intersections along with path optimization by avoiding congestion en route. This will result in reducing the cost of travel due to reduction in fuel consumption as well as minimize health risks such as skin diseases or asthma or lung-related disorders or irritation to eyes due to engine emissions etc.



In the current scenario, vehicles have to wait for a long time in order to get their turn to cross the intersection due to either pre-timed signals or actuated signals with some predefined time calculation mechanism without considering actual vehicle density at that time. Also, normally all the drivers are inclined to choose the shortest path to reach the destination, which increases the congestion on that path due to large number of vehicles using that path. To the best of our knowledge, in the literature, only one of the parameters is optimized to reduce overall trip time, i.e., either the problem of traffic signals or the routing algorithm is handled. In this paper, both of these parameters are optimized simultaneously by the proposed MACO-P algorithm for reducing trip time. In the proposed algorithm, MACO algorithm has been improved by incorporating preemptive traffic signals in place of normal traffic signals that come on the way to the destination. After implementing the MACO-P algorithm, a commuter is able to minimize the waiting time at the intersections with non-congested optimized path resulting in less travel time, which further reduces the cost of traveling to reach destination due to reduction in fuel consumption.

### ***Proposed MACO-P algorithm***

The proposed MACO-P algorithm has been designed by improving the existing MACO algorithm. Several traffic signals are encountered on the path selected by the MACO algorithm. These traffic signals (i.e., fixed or adaptive signals) increase the overall waiting time in total travel time due to their normal working. For reducing the waiting time at signals, MACO-P, which incorporates the preemptive traffic signal algorithm into the MACO algorithm to reduce the overall trip time by minimizing the overall waiting time at the signal, is used. In the MACO-P algorithm, first a non-congested path with minimum pheromone is selected by the MACO algorithm and then on that path whenever a traffic signal is encountered, the preemptive traffic signal algorithm is applied to reduce the waiting time at the intersection. The path is a collection of roads from the starting node to the destination node. The roads are connected by intersections that may have traffic signals installed on them. Each road has three lanes, first one for left turn, second for going straight and third for right turn. For implementation, we have taken random initial positions with random destinations for the vehicles in the network considered. As the vehicle moves on a path, its present position changes consequently by using the adjacency matrix. The working of the proposed algorithm is discussed below.

The vehicle moves on the road by sensing both its current position and pheromone value to reach the destination. Initially a vehicle is somewhere on the road and its actual position is taken as the current position. If the current position is the same as the destination position, another vehicle is taken into



consideration till all vehicles reach the destination. If the current position of the vehicle lies between the nodes, i.e., vehicle is waiting on the road either due to signal or congestion, then the new position for that vehicle is the next node on the same lane of the road. Otherwise, from the next adjacent nodes for the current position, road with minimum pheromone value and having vehicles less than the threshold have been selected and that node is checked for being a traffic signal and corresponding road has been selected. If the selected road has traffic signals, preemptive algorithm is applied; otherwise present position of the vehicle is updated with the next node position for that vehicle. On the selected road the pheromone value is increased and other roads in the network have this pheromone value decreased. For the determination of congestion, threshold is taken and calculated experimentally. The value of threshold is compared with the number of vehicles in simulation. Flowchart of the proposed algorithm is given in Figures 1 and 2. The pseudocode for the algorithm is presented in Figure 3. Here, preemptive algorithm is called in case of a traffic signal encountered on the path selected by the MACO algorithm to further reduce the waiting time at the

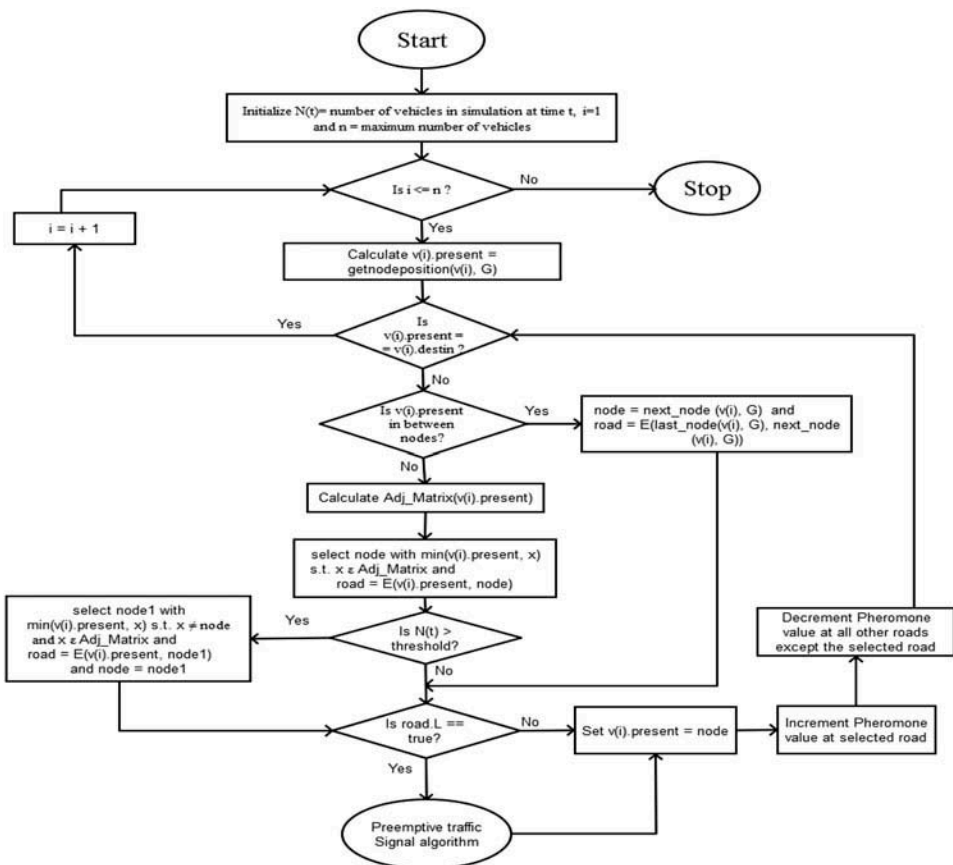


Figure 1. Flowchart for the MACO-P algorithm.

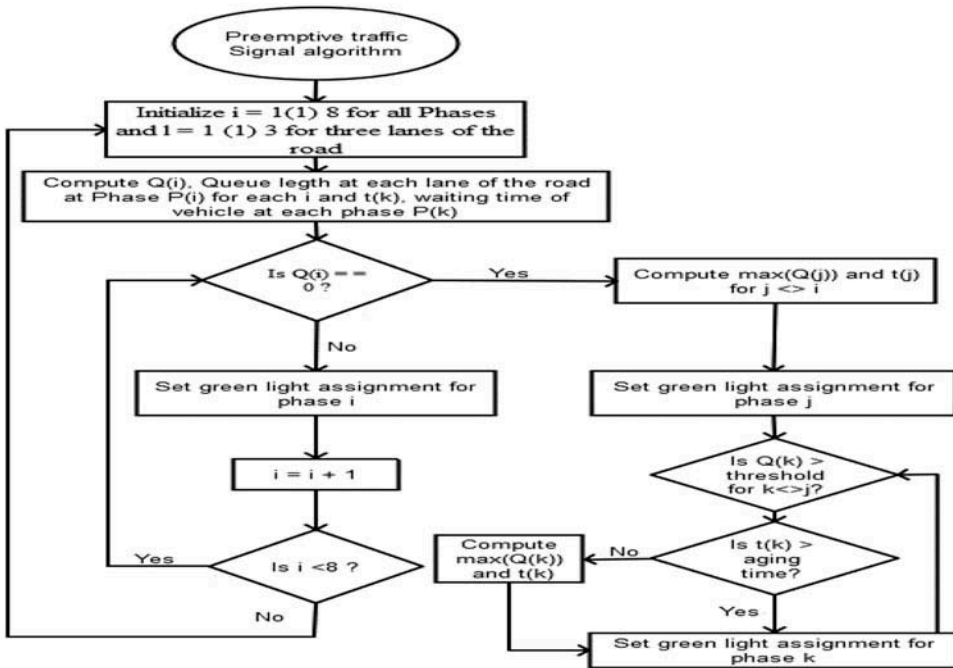


Figure 2. Pseudocode for the preemptive sub algorithm.

intersection. The proposed MACO-P algorithm like MACO algorithm works under the assumption that all roads are in normal working condition. The MACO-P algorithm is summarized below:

- If vehicle is in between nodes, next node on the same path is selected as next node.
- Otherwise next node with minimum pheromone value on the path from the current position to that node among the adjacent nodes is found.
- If the path between current node and next node has traffic signals, preemptive algorithm is used to handle the signals.
- Position of the vehicles is updated after handling the traffic signals and finding the next node.

## Experimental setup and results

In this paper, we designed four types of road networks, viz. simple, complex, Real-time DU network and Real-time NWD network using the open source simulator: SUMO along with MOVE. We use simulators to check the efficiency of the proposed MACO-P algorithm. For the presented work, the road network is simulated using open source tools: Simulation of Urban Mobility (SUMO) (DLR and contributors SUMO-Homepage 2014; Vaza, Parmar, and

**Pseudocode: Preemptive MACO (MACO-P) (L:Light, P:Phase)**

$G(N,E)$  = graph,  $G$  with node set,  $N$  and edge set,  $E$   
 $Ph_{r(a,b)}$  = value of pheromone on road,  $r$  from node  $a$  to node  $b$   
 $N_t$  = number of vehicles in simulation at time,  $t$   
 $Q_i$  = Queue Length at each lane of the road for phase,  $P_i$   
 $t_j$  = Waiting time of vehicle at phase,  $P_j$   
 for  $i = 1(1) n$   $r$

```

    vehi.present = getnodeposition(vehi,G)
    if vehi.present = vehi.destin
        i++
    else if vehi.present is in between the nodes
        node = next_node(vehi, G)
        road = E(last_node(vehi, G), next_node(vehi, G))
    else
        calculate A(vehi.present) //adjency matrix for vehi.present ∈ N
        select node with min (E(vehi.present, x)) s.t. x ∈ A(vehi.present)
        road = E(vehi.present, node)
        if Nt > threshold
            select node1 with min (E(vehi.present, x)) s.t. x ≠ node and x ∈ A(vehi.present)
            road = E(vehi.present, node1)
            node = node1
        end if
    if road.L == true //road has traffic signals
        for u = 1 (1) 8
            if Qu == 0
                for all values of v s.t. v ≠ u
                    if all Qv ≠ 0
                        compute max(Qv)
                        Lv = 1 for max(Qv) //Green light Assignment for phase, v
                    for all values of w s.t. w ≠ v
                        go to Label
                    end for
                else
                    Lu = 1 //Green light Assignment for phase, u
                    for all values of w s.t. w ≠ v
                        go to Label
                    end if
                Label:
                for any Qw
                    if Qw > Thmax
                        Lw = 1 //Green light Assignment for phase, w
                    else if tw > Tmax
                        Lw = 1 //Green light Assignment for phase, w
                    else
                        compute max(Qw)
                        Lw = 1 for max(Qw) //Green light Assignment for phase, w
                    end for
                continue for loop
            vehi.present = node
            Phroad++ //Increment pheromone at the selected road
            Ph[N-road]-- //Decrement pheromone at all other roads accept the selected road
        continue for loop
    
```

**Figure 3.** Pseudocode for the MACO-P algorithm.

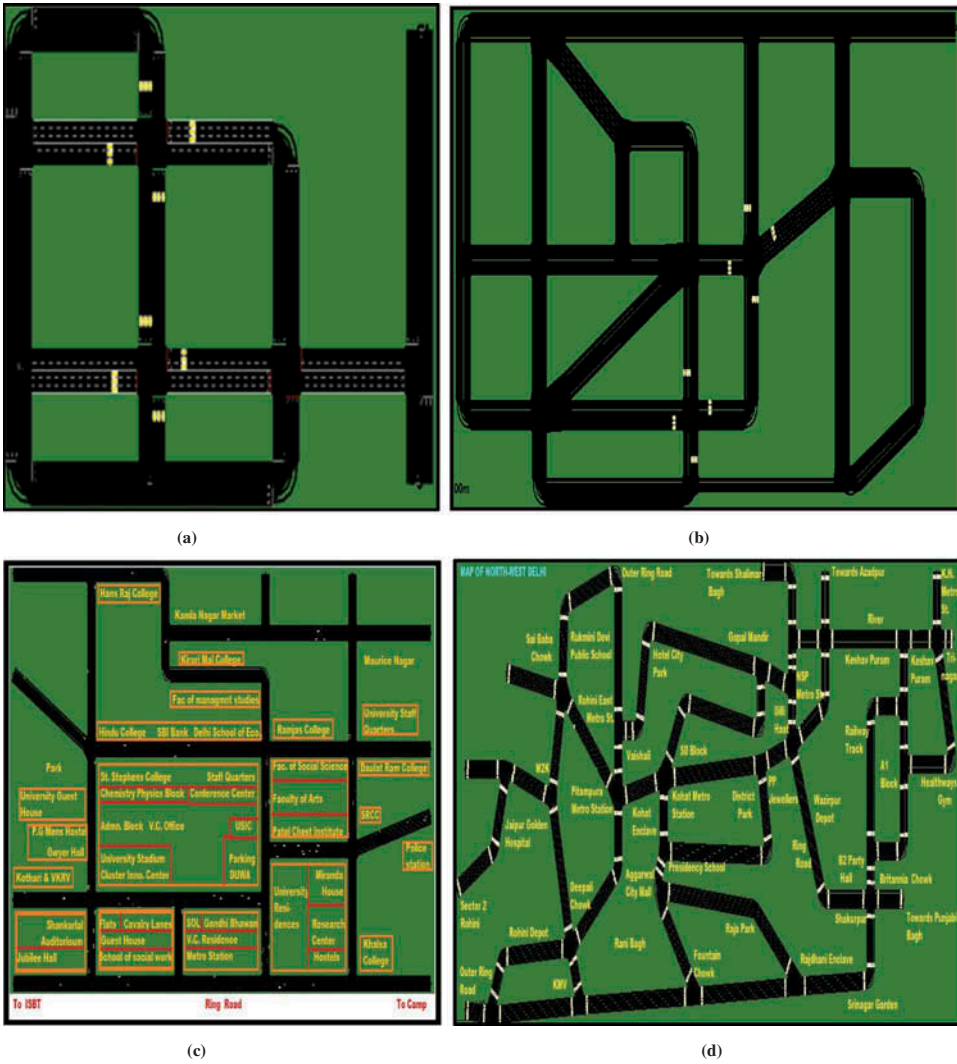
Kodinariya 2014), Traffic Control Interface (TraCI) (TraCI/Change 2008), Mobility model generator for Vehicular networks (MOVE) (Lan and Chou 2011) and Python for generating client scripts that start and control the complete simulation. MOVE, a Graphical User Interface (GUI) mobility model generator is used on top of SUMO, a micro-traffic simulator.

SUMO uses real-world vehicle information stored in a mobility trace file which is generated by MOVE (Khairnar and Pradhan 2010). The combination of SUMO and MOVE is known as traffic simulator and is used for modeling the vehicles' mobility and generating traffic information. This combination works as the server for the system.

TraCI is used as an extension of SUMO that allows it to communicate with external applications like python or java, where SUMO acts as server and external application acts as client to initiate simulation. Since SUMO does not have inherent sensor network modules TraCI helps in communication with such networks. It can perform command line-based simulations using GUI, import various types of maps, consider multi-lane roads and traffic signs and configure individual routes for vehicles. In the present work, we are using python scripts for controlling the communication. Out of the four road networks used in the paper, two are artificial networks: Simple network and Complex network. Other two are real-time simulated networks: DU network and NWD network. The snapshot of simple road network is shown in Figure 4(a). It consists of 14 nodes that are connected by 36 roads. Each road consists of 3 lanes. After the simple simulation network a complex network was designed and its snapshot is shown in Figure 4(b). The complex network consists of 22 nodes that are connected by 65 roads. Each road consists of 3 lanes. In the simulated networks, nodes are junctions in the road network and the streets connecting the junctions are the edges in the network. The cost for choosing the next best node is the length of the street. This resembles the real-life scenario as commuters follow the shortest path from the source to destination on roads. SUMO also allows us to modify the parameter for edge selection as it could also be the travel time taken on that particular edge.

After designing simple and complex networks, performance of the proposed algorithm was verified on a real-time DU network. Figure 4(c) represents the real-time DU network referred from the website of University of Delhi. This network consists of 29 nodes that are connected by 70 roads. Each road is of 3 lanes. Another real-time network on which the algorithms were tested, the NWD network, is shown in Figure 4(d). This network further verified the effectiveness and accuracy of the algorithms as this network is complicated and more closely imitates the real-world road scenario as compared to the other networks. The network consists of 52 nodes that are connected by 128 roads. Each road is of 3 lanes.

In the MACO-P algorithm, a random and uniform speed range of 7–12 m/s was used for the vehicle mobility. To test the efficiency of the algorithm, simulations were run 500 times with varied number of vehicles and average values have been recorded. For the selection of congestion-free path, MACO-P algorithm uses the MACO routing algorithm which optimizes the path length and testing was carried out with varied number of vehicles. During the journey, whenever a traffic signal comes in the path, preemptive traffic control



**Figure 4.** Snapshots of various networks used: (a) simple road network, (b) complex road network, (c) University of Delhi network, and (d) North West Delhi network.

algorithm has been applied. Obtained results are compared with both the MACO and Dijkstra’s algorithms. The MACO-P algorithm provides considerable decrement in the total travel time for the vehicles because of the reduction in waiting time. It was assumed that the network consists of too many vehicles with congestion on many of its routes. Vehicles follow the modified ACO path only when vehicles in the network exceed some set threshold, i.e., some congestion exists, otherwise the path followed by vehicles is the default path given by the Dijkstra’s algorithm.

The implementation of the algorithms was done in Python. Testing was performed over all the four simulated road networks and results were compared. The

two parameters on which comparisons are performed are overall travel time for reaching the destination and the overall waiting time in the entire route. The results justify that MACO-P outperforms the other two algorithms: Dijkstra's algorithm and MACO algorithm, when there is congestion in the network returning the optimal route that is obstruction free. Table 1 summarizes the results obtained by implementing the MACO-P algorithm along with two routing algorithms in all four simulated networks over the two parameters specified. Figure 5 and 6 shows the graphical results of overall waiting time and travel time in all the four networks. While waiting at the intersections, often the engine of the vehicles is kept on, which results in wastage of precious fuel. The proposed algorithm is also able to reduce fuel consumption by reducing the overall waiting time on both intersections as well by avoiding congested routes. Further, it was also inferred that the amount of the pollution emitted by the vehicles is less due to reduced waiting time on the roads. This also reduces various health hazards and leads to an eco-friendly atmosphere. Tested with all the four different networks on varied number of vehicles in each simulation, the MACO-P algorithm reduces congestion at intersections by handling heavy traffic.

### ***Statistical inferencing for the proposed MACO-P algorithm***

For analyzing the performance of the proposed MACO-P algorithm statistically with respect to other existing algorithms, the statistical hypothesis testing using t-test is chosen in the paper. The t-test is a kind of hypothesis test that checks the acceptance of the specified null hypothesis. The t-test with two paired samples for means assuming unequal variance is used in this paper. Null hypothesis ( $H_0$ ) and alternate hypothesis ( $H_1$ ) are defined in the paper as follows:

$$H_0 : \mu_1 \geq \mu_2.$$

and

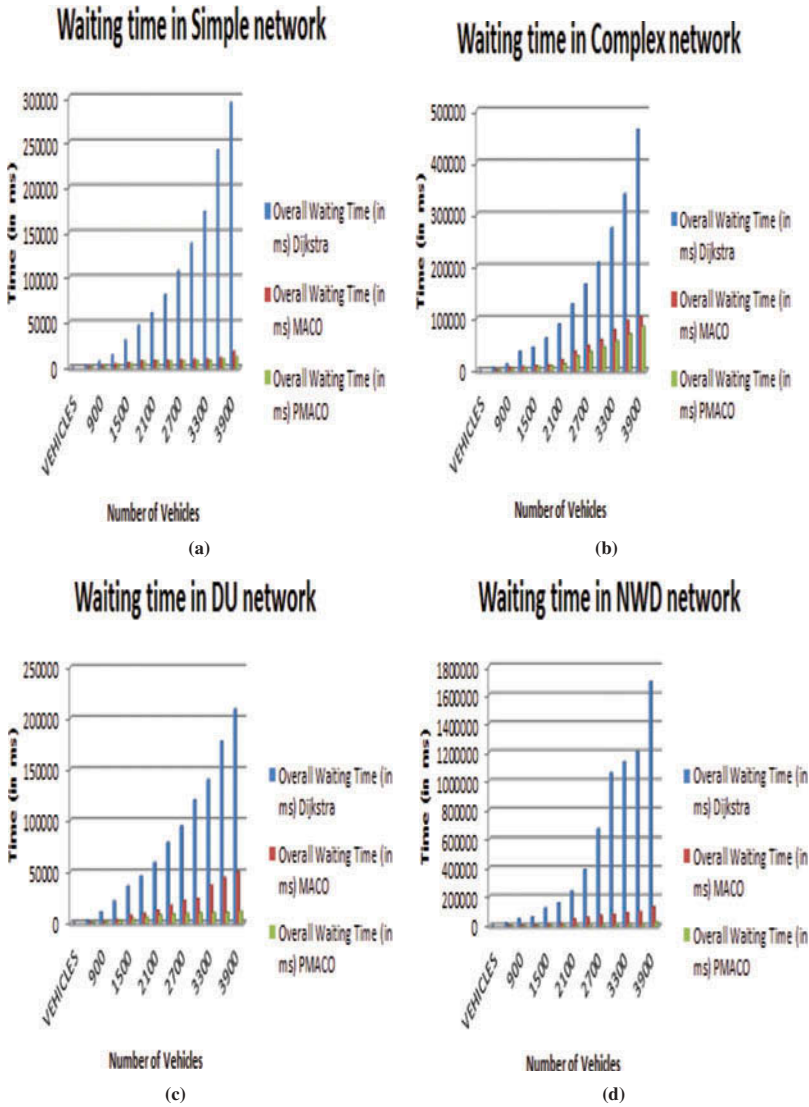
$$H_1 : \mu_1 < \mu_2.$$

where  $\mu_1$  is the mean of overall travel time in the first algorithm in consideration and  $\mu_2$  is the mean of overall travel time in the second algorithm under t-test. In significance testing, T is a test statistic used for the evaluation of the observed data that, whether it satisfies the null hypothesis or not. If the value for the observed test statistic varies considerably from the value for the hypothesized values, then null hypothesis is rejected and we claim the better performance of the algorithm under consideration, otherwise null hypothesis is accepted. The equation used to compute the t-statistic is given by Equation (1). Here  $\mu_1$  and  $\mu_2$  are the means,  $s_1$  and  $s_2$  are the standard deviations and  $n_1$  and  $n_2$  the sample sizes of the two algorithms.

**Table 1.** Waiting and travel times in various networks.

VEHICLES	Overall waiting time (in ms)				Overall travel time (in ms)				Overall waiting time (in ms)				Overall travel time (in ms)									
	MACO		MACO-P		Dijkstra		MACO		MACO-P		Dijkstra		MACO		MACO-P		Dijkstra		MACO		MACO-P	
	Dijkstra	MACO	MACO-P	MACO-P	Dijkstra	MACO	MACO	MACO-P	Dijkstra	MACO	MACO-P	MACO-P	Dijkstra	MACO	MACO	MACO-P	Dijkstra	MACO	MACO	MACO-P	MACO-P	
Simple network																						
600	2839	1577	1329	24,373	30,777	30,777	28,453	5018	1620	1582	35,188	35,635	34,954	7027	2108	1984	41,256	12,174	5742	3214	55,516	52,419
1200	14,111	4396	3654	53,655	61,551	61,551	55,214	36,538	7432	4568	116,326	95,094	78,546	30,182	5343	4487	75,412	44,597	9240	7542	117,838	88,672
1800	47,203	7777	6521	106,768	96,399	96,399	94,785	62,629	10,001	8659	151,291	122,942	92,457	2100	60,882	8138	130,059	89,117	20,617	12,405	192,514	126,836
2400	80,730	8451	7269	159,807	112,654	112,654	10,058	126,963	36,529	28,035	244,723	131,334	101,458	2700	107,544	8877	196,855	165,376	48,079	36,547	297,464	135,107
3000	137,573	9382	7862	236,929	122,767	122,767	10,859	207,776	59,270	45,025	354,539	143,862	10,987	3300	172,673	9698	281,866	274,113	78,108	56,457	434,487	169,021
3600	240,870	11,017	9504	509,167	126,888	126,888	11,028	338,134	95,566	69,542	562,966	171,275	11,231	3900	292,849	18,255	821,570	462,546	103,635	84,579	651,297	188,726
Real-time DU network																						
600	2878	529	342	46,789	42,967	42,967	40,215	18,618	1600	1325	90,051	74,162	65,478	900	10,975	1371	75,277	49,856	3729	1987	150,720	101,590
1200	21,624	3101	2541	107,066	101,207	101,207	65,489	61,750	7490	2354	198,286	143,385	85,479	1500	36,361	7286	140,261	120,676	11,677	4529	279,660	175,432
1800	45,967	9487	5487	173,127	170,757	170,757	96,587	158,252	15,051	6587	365,911	228,537	101,598	2100	59,248	12,209	208,321	243,077	48,684	7896	486,826	241,382
2400	78,841	17,264	8964	249,799	222,477	222,477	112,547	389,057	58,004	9648	667,761	289,037	112,689	2700	94,210	22,005	287,198	670,521	69,547	10,254	983,066	405,814
3000	120,114	23,922	9876	335,756	282,950	282,950	152,598	1,057,375	79,428	10,658	1,603,810	425,784	148,756	3300	140,041	36,941	377,792	1,131,682	88,751	11,025	2,413,606	454,785
3600	177,282	44,251	10,265	436,553	382,338	382,338	185,478	1,201,634	96,547	11,542	3,418,917	485,478	198,755	3900	209,047	50,279	490,118	1,692,484	131,524	25,487	4,545,012	504,125
Real-time NWD network																						
600	2878	529	342	46,789	42,967	42,967	40,215	18,618	1600	1325	90,051	74,162	65,478	900	10,975	1371	75,277	49,856	3729	1987	150,720	101,590
1200	21,624	3101	2541	107,066	101,207	101,207	65,489	61,750	7490	2354	198,286	143,385	85,479	1500	36,361	7286	140,261	120,676	11,677	4529	279,660	175,432
1800	45,967	9487	5487	173,127	170,757	170,757	96,587	158,252	15,051	6587	365,911	228,537	101,598	2100	59,248	12,209	208,321	243,077	48,684	7896	486,826	241,382
2400	78,841	17,264	8964	249,799	222,477	222,477	112,547	389,057	58,004	9648	667,761	289,037	112,689	2700	94,210	22,005	287,198	670,521	69,547	10,254	983,066	405,814
3000	120,114	23,922	9876	335,756	282,950	282,950	152,598	1,057,375	79,428	10,658	1,603,810	425,784	148,756	3300	140,041	36,941	377,792	1,131,682	88,751	11,025	2,413,606	454,785
3600	177,282	44,251	10,265	436,553	382,338	382,338	185,478	1,201,634	96,547	11,542	3,418,917	485,478	198,755	3900	209,047	50,279	490,118	1,692,484	131,524	25,487	4,545,012	504,125



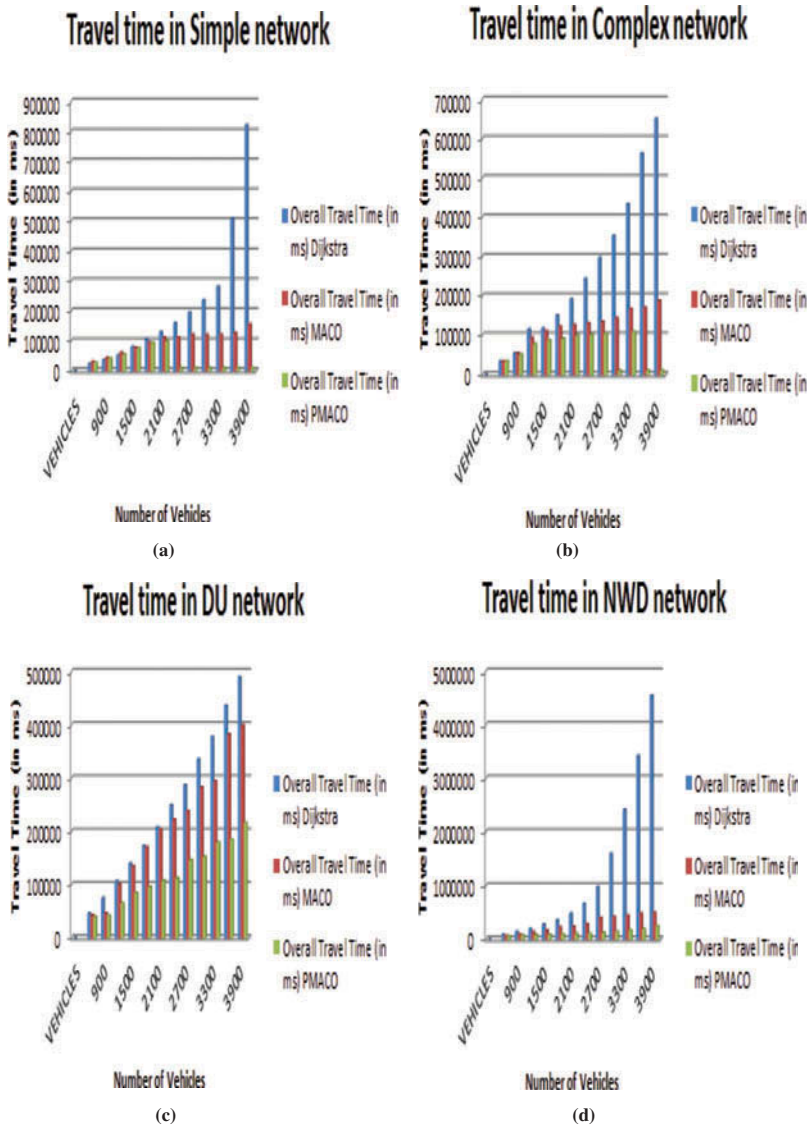


**Figure 5.** Overall waiting time in various networks: (a) simple road network, (b) complex road network, (c) University of Delhi network, and (d) North West Delhi network.

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (1)$$

### ***Inference using T-test for MACO-P algorithm***

For analyzing the performance of the proposed MACO-P algorithm over the standard Dijkstra's algorithm and existing MACO algorithm, the statistical hypothesis testing using t-test is performed in the paper. Out of these three algorithms, we have taken two algorithms at a time, i.e., first t-test for



**Figure 6.** Overall travel time in various networks: (a) simple road network, (b) complex road network, (c) University of Delhi network, and (d) North West Delhi network.

Dijkstra and MACO is taken and then MACO and MACO-P are selected for the test. The t-test performs a hypothesis test for the validity of null hypothesis on the parameter, overall travel time to prove the outperformance of the MACO-P algorithm over both the Dijkstra’s algorithm and the MACO algorithm. We have taken  $\mu_1$  as the mean of travel times for the first algorithm and  $\mu_2$  as the mean of travel times for the second algorithm under t-test. In the paper,  $\mu_1$  refers to the mean of travel times for the MACO algorithm or MACO-P algorithm and  $\mu_2$  refers to the mean of travel times for Dijkstra’s algorithm or MACO algorithm respectively. In

significance testing,  $T$ , a test statistic is used to calculate the belongingness of the observed data to the null hypothesis. If the values for the observed test statistic differ considerably from the values obtained for hypothesized values, then null hypothesis is rejected. It is inferred that the algorithm under examination is performing better than its counterpart, otherwise null hypothesis is accepted.

We have taken four different significance levels for  $\alpha$  equal to 0.04 0.06, 0.11 and 0.15 respectively for the investigation of t-test. The statistical hypothesis testing is done for all the four simulated networks (i.e., simple network, complex network, real-time DU network and real-time NWD network) pertaining to Dijkstra, MACO and MACO-P algorithms. To obtain the normalized values, experiments are done 500 times. In the statistical hypothesis, probability,  $p$  is considered for acceptance or rejection of the hypothesis. If the value of  $p$  is less than 0.04, then null hypothesis is rejected and alternate hypothesis is accepted with 96% confidence. Otherwise, the value of  $p$  is checked with another value of  $\alpha$ , null hypothesis is rejected in case of less value than that of 0.06 and alternate hypothesis is accepted with 94% confidence. Again, the value of  $p$  is checked again with the next value of  $\alpha$ , null hypothesis is rejected in case of less value than that of 0.11 and alternate hypothesis is accepted with 89% confidence. Finally, value of  $p$  is checked again with next value of  $\alpha$ , null hypothesis is rejected in case of less value than that of 0.15 and alternate hypothesis is accepted with 85% confidence. The results obtained for the t-test are recorded in [Table 2](#).

It is clearly inferred from [Table 2](#) that the MACO-P algorithm performs better than the MACO algorithm as the null hypothesis has been rejected and alternate hypothesis has been accepted with 96% confidence in all the four networks used in simulation. Further the MACO algorithm also performs better than Dijkstra's algorithm as the null hypothesis has been rejected and alternate hypothesis has been accepted with 96% confidence in all the four networks used in simulation. Hence, it is concluded that the performance of the MACO-P approach is better than both the Dijkstra and the MACO algorithms with the increase in the complexity of the network and traffic, which is the need of the current scenario, i.e., to cater to the increased and dynamic traffic demand in VANETs.

## Conclusion

A MACO-P algorithm for reducing the total travel time in VANETs is presented in this paper. Travel time in VANETs can be reduced by optimizing two factors: providing green time at traffic signals and choosing the optimal congestion-free route for the vehicles. MACO-P algorithm has been designed by improving the MACO algorithm with the inclusion of preemptive algorithm at the intersections. This combination provides

**Table 2.** t-test for Dijkstra, MACO and MACO-P algorithms in various networks in terms of travel time.

t-test: simple network			
	Dijkstra	Values for Dijkstra and MACO	MACO and MACO-P
Mean	219853		MACO 98814.67
Variance	53973616385.45		MACO-P 38307.75
Observations	12		1208297097.11
Hypothesized mean difference	0		12
df			0
t Stat		<b>12</b>	<b>22</b>
P(T ≤ t) one-tail		1.78	
t critical one-tail		<b>0.0301</b>	
		1.78	
t-test: complex network			
	Dijkstra	Values for Dijkstra and MACO	MACO and MACO-P
Mean	152081.75		MACO-P 29846.25
Variance	20716023940.75		814167678.02
Observations	12		12
Hypothesized mean difference	0		0
df			
t Stat	2.62		21
P(T ≤ t) one-tail		<b>0.0237</b>	
t critical one-tail		1.78	1.72
t-test: real-time DU network			
	Dijkstra	Values for Dijkstra and MACO	MACO and MACO-P
Mean	244004.75		MACO-P 119370.17
Variance	20821653524.02		3292760242.2
Observations	12		12
Hypothesized mean difference	0		0
df			
t Stat		21	16
P(T ≤ t) one-tail		0.63	2.41
t critical one-tail		<b>0.0268</b>	<b>0.0144</b>
t-test: real-time NWD network		1.72	1.75

(Continued)



Table 2. (Continued).

	Dijkstra		Values for Dijkstra and MACO		Values for MACO and MACO-P	
Mean	1266968.		MACO	294125.92	MACO-P	129304.08
Variance	2128443123525.8			24155422925.6		3141138985.36
Observations	12			12		12
Hypothesized mean difference	0			0		0
df		11				14
t Stat		2.29				3.46
P(T ≤ t) one-tail		<b>0.0211</b>				<b>0.0019</b>
t critical one-tail		1.79				1.76

both the optimization of waiting time at the intersection by reducing the average queue length using preemptive algorithm and optimization of the path length chosen for the vehicles using the pheromones to avoid congestion in the network by MACO algorithm. The MACO-P algorithm has the ability to pick any phase for the allotment of green time using the current vehicle density with the avoidance of starvation at any of the phases. Further, MACO-P algorithm is able to optimize the path length of vehicles and ensure the avoidance of the en route congestion by imitating the ant behavior.

The accuracy of the MACO-P algorithm was validated by comparing its results with the existing MACO algorithm and standard Dijkstra's algorithm. The MACO-P algorithm shows a decrement in both the total travel time taken by vehicles as well as the overall waiting time at intersections. The results also conclude that when vehicles in the network are less, then there is no significant improvement in the overall travel times in the MACO-P algorithm but as the number of vehicles increases there will be congestion, the travel times show a remarkable improvement as compared to that of the existing algorithms. Both experimental and statistical results show a significant decrease in total travel time as compared with its counterparts. The proposed MACO-P algorithm also reduces fuel consumption significantly as both waiting period for crossover at intersection will be minimized and at the same time it reduces the total journey time as congested stretch will be avoided. Further, the amount of various pollutants emitted by the vehicles will be considerably less due to the factors mentioned above, thereby leading to an environment-friendly atmosphere which will ensure least health hazards for the persons in the area concerned.

## ORCID

Vinita Jindal  <http://orcid.org/0000-0002-0481-4840>

Punam Bedi  <http://orcid.org/0000-0002-6007-7961>

## References

- Al-Sultan, S., M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan. 2014. A comprehensive survey on vehicular Ad Hoc network. *Journal of Network and Computer Applications* 37:380–92. doi:10.1016/j.jnca.2013.02.036.
- Altayeb, M., and I. Mahgoub. 2013. A survey of vehicular Ad hoc networks routing protocols. *International Journal of Innovation and Applied Studies* 3 (3):829–46.
- Bedi, P., V. Jindal, H. Dhankani, and R. Garg. 2015. ATsOT: Adaptive traffic signal using mOTes. Proceedings of the 10th International Workshop on Databases in Networked Information Systems (DNIS'15), Aizu-Wakamatsu, Japan, Springer International Publishing, Switzerland, 152–171.

- Bedi, P., V. Jindal, R. Garg, and H. Dhankani. 2015. 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.
- Bedi, P., N. Mediratta, S. Dhand, R. Sharma, and A. Singhal. 2007. Avoiding traffic jam using ant colony optimization—A novel approach. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications. Sivakasi, India, IEEE, 61–67.
- 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.
- Bin, Y., Y. Z. Zhen, and Y. Baozhen. 2009. An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research* 196 (1):171–76. doi:10.1016/j.ejor.2008.02.028.
- Cabanes, G., E. Van Wilgenburg, M. Beekman, and T. Latty. 2015. Ants build transportation networks that optimize cost and efficiency at the expense of robustness. *Behavioral Ecology* 26 (1):1–9.
- Chaqfeh, M., A. Lakas, and I. Jawhar. 2014. A survey on data dissemination in vehicular ad hoc networks. *Vehicular Communications, Elsevier* 1 (4):214–25. doi:10.1016/j.vehcom.2014.09.001.
- Claes, R., and T. Holvoet. 2011. Ant colony optimization applied to route planning using link travel time prediction. International Symposium on Parallel Distributed Processing, IEEE, 358–65.
- Dempsey, P., and A. Schuster. 2005. Swarm intelligence for network routing optimization. *Journal of Telecommunications and Information Technology* 3:24–28.
- Dhankhar, S., and S. Agrawal. 2014. VANETs: A survey on routing protocols and issues. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)* 3 (6):13427–35.
- Dias, J. C., P. Machado, D. C. Silva, and P. H. Abreu. 2014. An inverted ant colony optimization approach to traffic. *Engineering Applications of Artificial Intelligence* 36:122–33. doi:10.1016/j.engappai.2014.07.005.
- DLR and contributors SUMO-Homepage. 2002. [http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/) (accessed October 13, 2014).
- Dorigo, M., G. D. Caro, and L. M. Gambardella. 1999. Ant algorithms for discrete optimization. *Artificial Life* 5 (2):137–72. doi:10.1162/106454699568728.
- Dorigo, M., and S. Thomas. 2004. *Ant colony optimization*. Cambridge, Massachusetts, USA: MIT Press.
- Dua, A., N. Kumar, and S. Bawa. 2014. A systematic review on routing protocols for vehicular Ad Hoc networks. *Vehicular Communications, Elsevier* 1 (1):33–52. doi:10.1016/j.vehcom.2014.01.001.
- Elloumi, W., H. E. Abed, A. Abraham, and A. M. Alimi. 2014. 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.
- Falocchio, J. C., and H. S. Levinson. 2015. *Road traffic congestion: A concise guide*, vol. 7. Switzerland: Springer International Publishing.
- Gartner, N. H. 1982. *Demand-responsive decentralised urban traffic control*. US Department of Transportation, Research and Special Programs Administration, U.S. Department of Transportation, Lowell, USA: Office of University Research, U.S. Department of Transportation.
- Geetha, S., P. T. Vanathi, and G. Poonthalir. 2012. Metaheuristic approach for the multi-depot vehicle routing problem. *Applied Artificial Intelligence* 26 (9):878–901. doi:10.1080/08839514.2012.727344.



- Gradinescu, V., C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode. 2007. Adaptive traffic light using Car-to-Car communications. Proceedings of the IEEE 65th Vehicular Technology Conference (VTC), Dublin, Ireland: IEEE, 21–25.
- Ho, C. K., and H. T. Ewe. 2009. GACO—A hybrid ant colony optimization metaheuristic for the dynamic load-balanced clustering problem in Ad Hoc networks. *Applied Artificial Intelligence* 23 (7):570–98. doi:10.1080/08839510903161139.
- Jindal, V., H. Dhankani, R. Garg, and P. Bedi. 2015. 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.
- Kakarla, J., S. S. Sathya, B. G. Laxmi, and B. B. Ramesh. 2011. A Survey on Routing Protocols and its Issues in VANET. *International Journal of Computer Applications* 28 (4):38–44. doi:10.5120/3373-4663.
- Khairnar, V. D., and S. N. Pradhan. 2010. Mobility models for vehicular Ad-hoc network simulation. *International Journal of Computer Applications* 11 (4):8–12. doi:10.5120/1573-2103.
- Khekare, G. S., and A. V. Sakhare. 2012. Intelligent traffic system for VANET: A survey. *International Journal of Advanced Computer Research* 2 (4):99–102.
- Lan, K. C., and C. M. Chou. 2011. An open source vehicular network simulator. Proceedings of OSDOC'11, Lisboa, Portugal: ACM, 1–2.
- Lee, K. C., U. Lee, and M. Gerla. 2010. Survey of routing protocols in vehicular ad hoc networks. In *Advances in vehicular ad-hoc networks: Developments and challenges*, M. Watfa, Ed., 149–70. USA: Idea Group Inc (IGI) Global.
- Li, F., and Y. Wang. 2007. Routing in vehicular Ad Hoc Networks: A survey. *Vehicular Technology Magazine, IEEE* 2 (2):12–22. doi:10.1109/MVT.2007.912927.
- Liang, W., L. Zhuorong, H. Zhang, S. Wang, and R. Bie. 2014. Vehicular Ad Hoc Networks: Architectures, research issues, methodologies, challenges, and trends. *International Journal of Distributed Sensor Networks* 11 (8):17–27.
- Lowrie, P. R. 1982. The sydney co-ordinated adaptive traffic system—principles, methodology, algorithms. Proceedings of the International Conference on Road Traffic Signaling. London, United Kingdom, 67–70.
- Maslekar, N., M. Boussedjra, J. Mouzna, and H. Labiod. 2011. VANET based adaptive traffic signal control. Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC Spring), Budapest, Hungary:IEEE, 1–5.
- Mimbela, L. E. Y., and L. A. Klein. 2000. *Summary of vehicle detection and surveillance technologies use in intelligent transportation systems*. Intelligent Transportation Systems Joint Program Office, Federal Highway Administration, Washington DC, USA: Joint Program Office for Intelligent Transportation Systems.
- Mirchandani, P., and L. Head. 2001. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9 (6):415–32. doi:10.1016/S0968-090X(00)00047-4.
- Moustafa, H., and Y. Zhang. 2009. *Vehicular networks: Techniques, standards, and applications*. Boston, MA, USA: Auerbach publications.
- Nanda, B. K., and G. Das. 2011. Ant colony optimization: A computational intelligence technique. *International Journal of Computer Communication Technology* 2 (6):105–10.
- Pandit, K., H. Dipak Ghosal, M. Zhang, and C.-N. Chuah. 2013. Adaptive traffic signal control with vehicular Ad hoc networks. *IEEE Transactions on Vehicular Technology* 62 (4):1459–71. doi:10.1109/TVT.2013.2241460.
- Piran, M. J., G. Rama Murthy, and G. Praveen Babu. 2011. 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.

- Rana, S., S. Rana, and K. C. Purohit. 2014. A review of various routing protocols in VANET. *International Journal of Computer Applications* 96 (18):28–35. doi:10.5120/16896-6946.
- Rehman, S. U., M. A. Khan, T. A. Zia, and L. Zheng. 2013. Vehicular Ad-Hoc Networks (VANETs)-An overview and challenges. *Journal of Wireless Networking and Communications* 3 (3):29–38.
- Roess, R. P., E. S. Prassas, and W. R. McShane. 2011. *Traffic engineering*, 4th ed. New York, USA: Pearson Higher Education.
- Shivani, Shivani, and J. Singh. 2013. Route planning in vanet by comparative study of algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering* 3 (7):682–89.
- Syal, P., and T. Kaur. 2014. A study of routing protocols for vehicular Ad-Hoc networks. *International Journal of Engineering Trends and Technology* 15 (1):25–29. doi:10.14445/22315381/IJETT-V15P206.
- TraCI/Change Traffic Lights State. 2008. DLR - Institute of Transportation Systems-SUMO-Simulation of Urban MObility. [http://sumo.dlr.de/doc/current/docs/userdoc/TraCI/Change\\_Traffic\\_Lights\\_State.html](http://sumo.dlr.de/doc/current/docs/userdoc/TraCI/Change_Traffic_Lights_State.html) (accessed October 4, 2014).
- Turner-Fairbank Highway Research Center. 2005. *Adaptive control software*. Report, Federal Highway Administration, Washington DC, USA: Federal Highway Administration, USA.
- U.S. Department of Transportation. 2004. Intelligent Transportation System (ITS). <http://www.its.dot.gov/index.htm> (accessed January 25, 2014).
- Vasirani, M., and S. Ossowski. 2011. Learning and coordination for autonomous intersection control. *Applied Artificial Intelligence* 25 (3):193–216. doi:10.1080/08839514.2011.551318.
- Vaza, R. N., A. B. Parmar, and T. M. Kodinariya. 2014. Implementing current traffic signal control scenario in VANET using SUMO. Proceedings of the National Conference on Emerging Trends in Computer and Electrical Engineering (ETCEE'14), India, 112–16.