Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 19, Iss. 2, pp. 15-28, 2021

A BENCHMARK FOR MULTI-OBJECTIVE ROUTING IN VEHICLE AD-HOC NETWORKS USING THE ANT COLONY OPTIMIZATION ALGORITHM

Rodrigo Silva , **Heitor Silvério Lopes**
Federal University of Technology – Paraná (UTFPR), Curitiba, Brazil
rodrigo_silvabr@yahoo.com, hslopes@utfpr.edu.br

Abstract – The growing number of vehicles in cities has a great impact on our quality of life, such as air and noise pollution, traffic jams and traffic accidents. Cooperative Intelligent Transportation System (C-ITS) relies on communication technologies to provide innovative services and applications for transportation and traffic management. In the C-ITS context, users, roadside infrastructure and vehicles need to be connected and, for this purpose, a wide variety of wireless technologies can be used (e.g, vehicular WiFi, cellular and visible light communication). In this work we consider a VANET (Vehicular Ad-hoc NETwork) using vehicular WiFi (based on 802.11p). The communications in VANET networks have been studied for years and several routing algorithms have been developed for such a kind of network. However, a benchmark to compare the performance of such algorithms is still lacking. To fill this gap, the present work proposes a benchmark composed by instances of data routing for different scenarios in the VANET. Moreover, we propose a multi-objective algorithm based on ACO (Ant Colony Optimization) to compare with such benchmark. The results of simulations show the impact of several factors in the VANET connectivity, such as vehicle density, geographical location, propagation and fading models. The results are promising and indicate the importance of choosing appropriated simulation models.

Keywords – Ant Colony Optimization, C-ITS, Vehicular Ad-hoc Network, VANET, Benchmark, NS-3.

1 Introduction

The last data available from the International Organization of Motor Vehicle Manufacturers pointed out 1,282,270,000 vehicles in use in 2015 [1]. Since the number of vehicles on the planet is growing fast, it is estimated that there could be up to 2 billion vehicles circulating by 2035 [2]. Such growth has a great impact on our quality of life, specially in developed countries. For instance, the larger the number of vehicles, the more road traffic, air and noise pollution, traffic jam and traffic accidents.

Cooperative Intelligent Transportation System (C-ITS) provide solutions for a smarter use of transportation networks. C-ITS applies information and communication technologies for providing innovative services and applications related to transportation and traffic management. In the C-ITS context, users, roadside infrastructure and vehicles need to be connected anywhere, anytime, and with anything.

International standardization organisms (e.g., ISO and ETSI) have worked to a convergent ITS architecture that aims at making these wide variety of communicating devices interoperable [3]. Many technologies for the interconnection of C-ITS devices are available nowadays. For instance: vehicular WiFi [1], 802.15.4, cellular (3G, 4G, and 5G) and VLC (Visible Light Communication).

In this work we consider a VANET (Vehicular Ad-hoc NETwork) using vehicular WiFi. VANET is a subgroup of Mobile Ad-hoc Network (MANET), and they were first introduced by [4]. Until 2015, VANETs were understood for intra-vehicular communication. However, since then other type of communications started to be explored, between vehicles and the roadside infrastructure (traffic lights, buildings, antennas, etc.). Therefore, in such networks the communication can be Vehicle-to-Vehicle (V2V) and Vehicle-to-Roadside Infrastructure (V2I). Consequently, given the dynamics of a transportation system, VANETs are, at the same time, architectural networks (when considering V2I communication) and ad-hoc networks (considering V2V communication).The final objective is not only to provide user-focused services (such as internet connectivity) but, also, to improve the overall flow and security of the transportation network. This can be accomplished by preventing collisions, avoiding traffic jams, blind crossing. Currently, there is a great interest for VANETs in the scope of smart-cities [5].

Due to the unique features of VANETs, highly dynamic topology, large number of nodes, sparse connectivity in specific regions, and very small number of static routers, optimization methods are essential for an efficient operation of real-world systems, particularly routing [6, 7] and clustering [8] methods. Communications in the context of VANETs have been widely studied [9–11], and variety of routing algorithms have been developed [12, 13]. Amongst them, bio-inspired algorithms can be highlighted [14], including the Ant Colony Optimization (ACO) [15–17].

The ACO is an algorithm based on the ants' foraging behavior [15]. Ant colonies are capable of self-organizing into groups and develop complex tasks. Several works have shown that ants tend to navigate through the shortest path between their colony and the food source [18]. This behavior is explored in optimizations problems that can be represented by a graph, and a possible solution is a specific path in such a graph. Currently, there are many applications of ACO to real-world problems, such as: vehicle routing problems [19], bioinformatics [20], image processing [21], and data mining [22], to cite a few.

[1] ITS-G5 in Europe and DSRC in North America, both based on 802.11p

Despite the wide variety of VANET's routing algorithms in the literature, a benchmark to compare the performance of such algorithms is still lacking. To fill this gap, the present work proposes a benchmark composed by instances of data routing for different scenarios in VANET.

The paper is organized as follows. Section 2 overviews the most relevant algorithms based on ACO for routing in VANET. Section 3 depicts the problem approach. The simulation methodology is given in Section 4. Section 5 presents the configuration of the experiments for each scenario composing the benchmark. The simulations' results and their analysis are described in Section 6. Finally, Section 7 concludes the paper and proposes some future works.

2 Related Work

Few ACO-based routing algorithms were developped for VANET. [23] proposed a reactive routing protocol named MAV-AODV. This protocol discovers routes of a VANET only when vehicles need to send data. In this algorithm, vehicles exchange beacon messages in order to gather information about neighbor vehicles' mobility. Then, each vehicle predicts the geographical position of other vehicles in their vicinity. With such information, it is possible to evaluate the lifetime of links. MAV-AODV uses the link's lifetime and the number of hops to choose a suitable path for data routing.

The hybrid routing protocol named MAZACORNET (*Mobility Aware Zone based Ant Colony Optimization Routing for VANET*) was introduced by [17]. According to the vehicle's speed and their movement pattern, this protocol separates the vehicles in clusters of interconnected zones. Into each zone, a proactive route discovering process is used, i.e., routes are updated periodically. To find routes between zones, a reactive route discovering process is performed. Based on the Euclidean distance between two vehicles, such protocol calculates the probability of a successful message exchange. With this information and the link stability estimation, the protocol determines the attractiveness of each path.

The Ant Route Search (AntRS) [16] is a proactive algorithm protocol based on the *AntSensor* protocol [24]. This algorithm considers that the probability of a connection between two vehicles is directly proportional to the Received Signal Strength (RSS), i.e., the link with greatest RSS is considered the one with less disconnection probability. From the Friis equation [25] it concludes that the received signal power between two vehicles is inversely proportional to square of the distance between them. Therefore, the link with smallest Euclidean distance between vehicles is the one with higher RSS and, consequently, the one with less disconnection probability. The AntRS applies the number of hops and Euclidean distance to determine the attractiveness of a given path.

Besides the above-mentioned approaches, other metaheuristic methods were already proposed for different aspects of VANET routing, such as Moth-Flame Optimization (MFO) [14], Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC) [26].

In general, ACO-based routing protocols use different variables to determine a routing path. Figure 1 is based on thorough research and it shows the main variables used by some recent research works [27]. The two variables most frequently used (in 58% of the published works) are the *Transmission Time* to send packets and the *Number of hops* in the path.

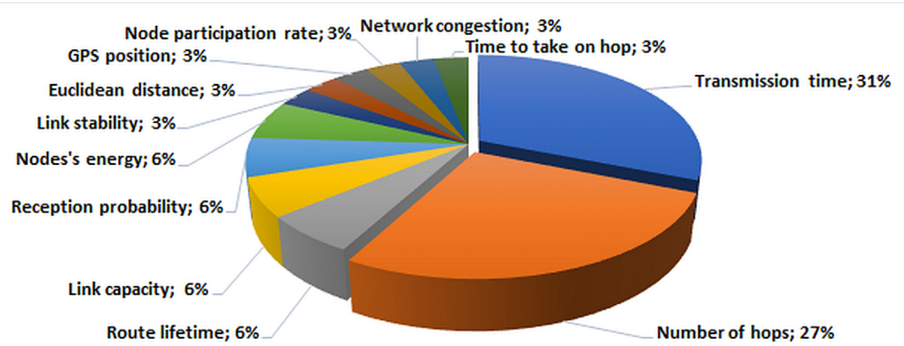

Figure 1: The most commonly used variables in ACO-based routing algorithm. Values refer to the frequency use in the recent literature [27].

3 Problem Description

Due to wide variety of protocols developed for data routing in VANETs and the lack of a benchmark to compare them, the present work aims at proposing a set of instances for routing in VANET networks. This proposed benchmark enables the comparison of routing protocols in a standardized way.

The basic problem focused here is defined as follows: given a VANET, find the best path to send data between two vehicles. Since there is a wide variety of issues that impacts data routing, the first step must be to choose the variables that define the best path. Based on the main variables used by ACO routing algorithms in the literature (as shown in Figure 1), we chose the two most relevant variables, i.e., *transmission time* and *number of hops*. Therefore, the best path is defined as the one with the shortest transmission time and lowest number of hops. Moreover, in the present work we propose a multi-objective algorithm based on the ACO, in order to compare with the newly proposed benchmark.

4 Methods

In order to generate each instance of the benchmark, several simulations were performed. Figure 2 shows an overview of the environment for such simulations. Each module is explained below.

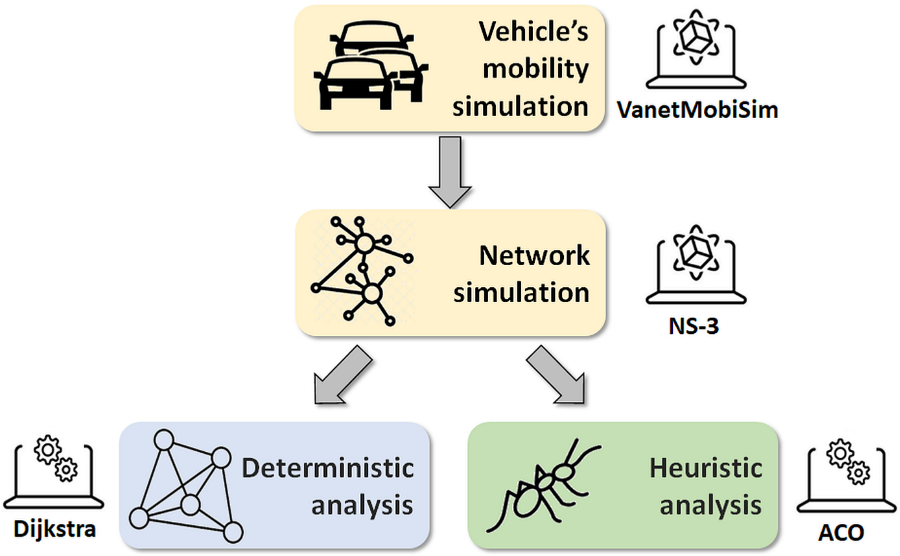

Figure 2: Overview of the simulation modules.

4.1 Vehicles' mobility simulation

The *VanetMobiSim* [28] simulator was used to perform simulations of the vehicles' mobility. *VanetMobiSim* is a Java-based application and it is capable of dealing with Geographical Data Files (GDF) and other standards. The simulator has several information levels that includes the roads topology, street particularities and traffic signs, and car constraints. It provides realistic mobility models for vehicular dynamics and includes V2V and V2I interactions at the microscopic level. *VanetMobiSim* outputs individualized trace files for each vehicle along the simulation, and these files are, then, used as input to the network simulations.

In this simulator several parameters can be configured, such as the simulation area, number of vehicles, mobility model, and others. The parameters used in our simulations are shown in Table 2 (Section 5).

4.2 Network simulation

The *Network Simulator 3* (NS-3) [29] is an open-source software aimed at simulating discrete-event networks. In this work, it was used to generate data traffic on the VANET previously created and to measure the transmission data time to send data packets between neighbors vehicles. Similarly to [28], NS-3 also has several parameters to be configured, such as routing protocol, access technology of wireless network (802.11p), and propagation model. The values for such parameters are shown in Table 2.

The *Flow Monitor* framework was used to monitor data traffic over the VANET. *Flow Monitor* is a framework developed for NS-3 that enables monitoring flows over each node in the network (e.g., packets sent, received or lost packets and time transmission) [30]. *Flow Monitor* allows a deep analysis of the network at each time stamp of the simulation.

The output of such simulation are files representing the network at each instant of time. They contain the neighbors list and the average measured time to send message between pairs of neighboring vehicles.

4.3 Deterministic analysis

To create a benchmark for data routing in a given network it is necessary to know the best path over such a network. The Dijkstra algorithm [31] is a deterministic procedure that is capable of finding the shortest path between nodes in of a graph. In this simulation, the Dijkstra algorithm receives the output of the NS-3 simulation and creates a graph $G(V, A)$ that represents the VANET. This graph is composed by V vehicles (vertices) and E connection links (edges) between neighbor vehicles. Each edge has a cost $C(Hops, Time)$, where $Hops$ is the number of hops and $Time$ is the average time to send a data packet between two connected vehicles.

Algorithm 1 describes in details the pseudocode of the Dijkstra algorithm used. The main loop begins in the vehicle source (V_i) and mark this node as visited. All others nodes are marked as unvisited with a infinite cost. At each step, the algorithm chooses an adjacent node with the lowest cost, relative to the initial node (V_i). Then, such a node (representing a vehicle) is added to a set S of the lowest cost path, and it is labeled as visited. Edges belonging to vehicles already in S are ignored. The algorithm is executed until all connected vehicles are visited.

Algorithm 1: Dijkstra algorithm

```
// Initialization
G(V, E) ← NS-3
V_i ← Source vehicle
V_f ← Destination vehicle
N ← Number of vehicles in the VANET
for j = 0 → (N − 1) do
    C_{j,i} ← ∞
    V_j ← unvisited
end
CurrentVehicle ← V_i
V_i ← visited
// Main loop
while ((G(V, E) ⊃ unvisited vehicles) & (V_f not yet visited)) do
    for k = CurrentVehicle → neighbors do
        Cost ← Cost of V_k related to V_i (Equation 1)
        if Cost < C_{k,i} then
            C_{k,i} ← Cost
            NextVehicle ← k
        end
    end
    CurrentVehicle ← NextVehicle
    V_currentvehicle ← visited
end
```

4.4 Objective function

Since this work proposes the use of a multiobjective function to be optimized, the weighted-sum method was used to combine the two objectives: *Number of hops* and *Transmission time* [32]. Considering a generic case, $\sum_i^N weight_i = 1$. For our specific case, it is assumed that both objectives have the same importance (weight) in the objective function. That is, each weight is set to 0.5. Depending on the situation, weights can be adjusted to reflect user's preferences regarding different scenarios.

Figure 3 depicts the objective space S (space of possible solutions) for two generic objective functions $f_1(x)$ and $f_2(x)$. C is the origin of the objective space, points B and D are the values of functions $f_1(x)$ and $f_2(x)$, respectively. The projection of B and D over the concave Pareto front of the solutions' space (red dotted line) leads to point A, which is the best compromise between functions $f_1(x)$ and $f_2(x)$ (considering their minimization).

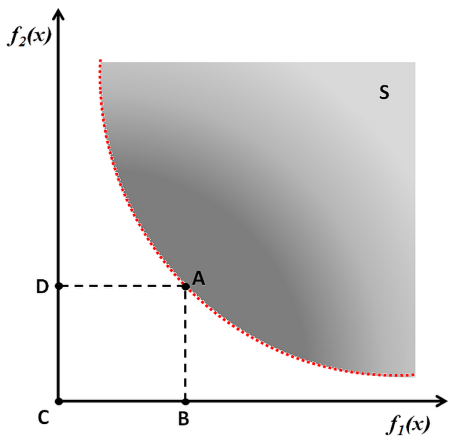

Figure 3: Objective space of $f_1(x)$ and $f_2(x)$ functions.

It is observed in Figure 3 that, the closer solution A is to the origin C, the better is the compromise of mutual minimization between the objectives. Therefore, the objective becomes the minimization of the line segment $\overline{AC}$ between origin of the objective space and the solution in S, i.e., minimization of the Euclidean distance between points A and C, given by: $d_{\overline{AC}} = \sqrt{(\overline{AB})^2 + (\overline{BC})^2}$. Replacing functions $f_1(x)$ and $f_2(x)$ by the functions *Number of hops* and *Transmission time*, respectively, the objective function (F_{Obj}) is defined by Equation 1:

$$F_{Obj} = \sqrt{\left(w_H \cdot \frac{Nhops}{N-1}\right)^2 + \left(w_T \cdot \frac{T_i}{\sum_{i=1}^{N} T_i}\right)^2} \quad (1)$$

where, N is the number of vehicles in the VANET, w_H and w_T are the weights of the functions, $Nhops$ are the number of hops, and T_i is the transmission time.

4.5 Heuristic analysis

For the heuristic analysis, the algorithm receives data from the NS-3 simulator and generates the graph of the network at each step of the simulation. Then, it starts searching for an optimal path between vehicles' source and destination in the graph. The proposed algorithm, named Ant Route Search for VANET (AntRSV), is based in the Ant Colony Optimization (ACO) algorithm first proposed by [15]. The ACO is a populational algorithm with biological inspiration. Basically, it has two components: an undirected graph that represents multi-paths between two points (e.g. source and target points), and a number of agents (ants) capable of moving through the edges of the graph according certain rules. As the ants move through the graph's edges, they leave behind certain amount of pheromone which, eventually, can evaporate along time. However, when a given ant reaches a vertex of the graph, the decision to follow a given edge is biased by the amount of pheromone already found. Along time, depending on the initial parameters and the dynamics of the system, ants may converge to the shortest path between the source and destination vertices.

We used a specialized version of the ACO, suited for the problem in hand, as shown in Algorithm 2. It was based on a former approach known as AntRS [16].

Algorithm 2: AntRSV

```
// Initialization
G(V, A) ← NS-3
V_i ← Source vehicle
V_f ← Destination vehicle
N ← Number of vehicles
// Create ants
for j = 0 → Number Of Rounds do
    Pheromone initialization - Equation 2
    while counter < Number Of Repetitions do
        for k = 0 → (N − 1) do
            Move ant (Equation 3)
            if ant reached Destination vehicle (V_f) then
                ant ← ant in return
                ant return to source vehicle (V_i)
            end
            else
                Choose the next vehicle
                Send ant
            end
        end
        Pheromone update
        Pheromone evaporation - Equation 4
        counter = counter + 1
    end
end
```

The main steps of the AntRSV algorithm are:

a) Graph creation: Once the algorithm receives vehicles' traceability performed by the NS-3 simulator, it creates a full graph representing network.

b) Pheromone initialization: In the original ACO algorithm [15], the initial value of the pheromones at the edges of the graph impacts the convergence speed of the algorithm. If the initial amount of pheromone is too low, the solution is quickly biased by the first ants circulating in the edges. On the other hand, if the initial amount of pheromone is too high, the first iterations are lost, i.e., the pheromone deposited by the first ants are masked by the excess of initial pheromone values. Therefore, in this work, we follow [15], and we use the initial pheromone as given by Equation 2:

$$Pheromone_{initial} = \frac{m}{\rho \cdot C_{initial}} \tag{2}$$

where m is the number of ants, ρ is the evaporation coefficient and $C_{initial}$ is the cost of an initial path. Such initial path is found by sending one ant from the source vehicle to a destination one.

c) Sending ants: The probability ($P_{i,j}$) for an ant to choose a path between the nodes i and j is given by the Equation 3.

$$P_{i,j} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k \, \in \, V_i}[\tau_{ik}]^{\alpha}[\eta_{ij}]^{\beta}} \tag{3}$$

where τ_{ij} is the amount of pheromone between the nodes i and j; V_i are the neighbors of the node i; η_{ij} is a heuristic information given by the inverse of the objective function shown in Equation 1; and α and β are parameters that weight the importance of the pheromone and heuristic parts, respectively.

d) Returning ants: when an ant arrives to the destination vertex, it returns to the source depositing pheromones on the path. Such pheromone amount deposited on the path is given by Equation 4.

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \tag{4}$$

where $\tau_{i,j} = 1/F_{Obj}$ is the amount of pheromone deposited on the edge between nodes i and j, m is the number of ants and $\Delta\tau_{i,j}^{k}$ is the amount of pheromone deposited by the ant k on the edge (i, j), and ρ is the pheromone evaporation rate.

Since ants tend to converge for the best solution over time, such a process of sending and receiving ants should be performed several times to enable ants to converge toward the best solution. The AntRSV algorithm refers to it as "*Number Of Repetitions*". Moreover, AntRSV is an heuristic algorithm. Therefore, for statistical analysis the AntRSV run several times for each scenario (i.e., "*Number Of Rounds*").

The ACO parameters used in the simulation are shown in Table 1, and follows [15].

Table 1: Values of the ACO parameters used in the experiments.

Parameter	Value
α	1
β	2
ρ	0.5
Number of ants (m)	25
Number of rounds	10
Number of repetitions	300

5 Experiments

For the simulation we chose a square area of 360,000 m^2 located in downtown Curitiba (Brazil). This is an area of high density traffic, surrounded by many tall buildings and, therefore, it is ideal for a real-world simulation.

Using the *VanetMobiSim* simulator, we distributed vehicles in this area and applied the IDM-LC (Intelligent Driver Model with Lane Changes) mobility model. Such a model was chosen because it better represents vehicles in the real world, making smart management of vehicles at crossroads and vehicles' overtake.

Aiming at realistic simulations, in this work we used the *Nakagami-m* fading model and the *Three Log Distance* propagation model. Such a choice was based on [33], who suggested that not all models found in the NS-3 simulator are adequate for VANET simulations.

The *Nakagami-m* is a stochastic model based on the signal fading. Therefore, it is a more realistic model for simulations of wireless communications. The model simulates multiple sources interference to the communication signal that leads to its fading. The model has parameters that modulate the severity of fading due to multiple path propagation.

The *Three Log Distance* is a deterministic propagation model frequently used for wireless communications, since it simulates the propagation of a signal in the presence of obstacles, such as buildings and vehicles. This model supposes that the signal loss is exponentially proportional to the distance between the emitter and the receptor. Therefore, it applies different losses for different distance intervals. Actually three distances are considered, so that four propagation regions (or fields) are formed. For each field a different signal reduction is computed by the simulator [33].

For each scenario we performed simulations with different vehicles' densities ($vehicles/km^2$): 0.07, 0.14, 0.28, 0.56 and 1.11. For all scenarios the source and destination vehicles were set on fixed and opposite positions on the simulation area, while

Table 2: Common parameters configured for each scenario.

Simulators	Parameter	Value
VanetMobiSim	Area (m^2)	600 x 600
	Mobility model	IDM-LC
	Vehicles' speed (Km/h)	0 – 60
	Vehicles' acceleration (m/s^2)	0.6
	Deceleration (m/s^2)	0.9
	Simulation time (s)	200
	Vehicles density	0.07; 0.14; 0.28; 0.56 and 1.11
NS-3	Propagation model	*Three Log Distance*
	Fading model	Nakagami-m
	Size of data packets ($bytes$)	512
	Transmission rate (bps)	16,384
	Transmission power (dB)	16.0206
	RSS (dB)	-96
	Access network	802.11p
	Routing protocol	AODV
	Sampling time (s)	2

the other vehicles were able to change their speed in the range $[0..60]$ Km/h. Table 2 shows a list of parameters common to all scenarios.

For the simulation scenarios, besides the vehicle density, the configuration of the propagation and fading models were also varied. Table 3 shows the specific configurations for each scenario [2]. The four scenarios are described as follows:

a) Scenario 1: In this scenario vehicles are randomly positioned in the simulation area, following the normal distribution. Such scenario does not consider any fading model. The main objective of this scenario is to analyze the impact of vehicles' density in the VANET environment.

b) Scenario 2: In this scenario, we consider the same initial vehicles' position of Scenario 1. However, here we apply the *Nakagami-m* fading model and the *Three Log Distance* propagation model.

c) Scenario 3: Similarly to Scenario 1, here, no fading model was considered. However, we distribute 80% of vehicles on the avenues and 20% of vehicles on the secondary streets. Using such a distribution we aim to better represent the traffic in the real-world.

d) Scenario 4: Here we consider the same vehicles' distribution from Scenario 3 (i.e., 80/20), but we used both, the *Nakagami-m* fading model and the *Three Log Distance* propagation model, similarly to what was done in Scenario 2.

6 Results and Analysis

First of all, recall that the Dijkstra algorithm can find the best path for each scenario. Overall, Table 4 shows the numerical results of the simulations performed. It is shown that no possible path between source and destination was found for networks with 0.07 and 0.14 vehicles density. However, the method succeeded to find paths for networks with larger number of vehicles.

According to [34] and [35] the connectivity between nodes in a wireless network is proportional with the network density. This phenomenon can be observed in the Scenario 1, for which no paths were found for networks with low vehicle density (e.g., 0.07, 0.14 and 0.28 $vehicle/km^2$). Figure 4 shows the cost of paths found by both Dijkstra and ACO (AntRSV) algorithms for Scenario 1, using 0.56 as vehicles density. For the costs found by the ACO algorithm, we plotted the best value of each simulation, i.e., the lowest cost among the 300 cycles. In the Tables 5, 6, 7, 8 of the Annex the absolute and relatives values, as well as the average and the standard deviation are shown. The values of parameters used in such simulation represent vehicles in a real world, where vehicles are constantly moving. Due to such a high mobility, vehicles suffers of frequent disconnection. Therefore, in most of time no path was found.

Figure 4 shows that the Dijkstra algorithm found better paths than ACO, i.e., low cost paths. Actually, in the ACO algorithm, ants tend to converge for the best solution over time. In order to show the convergence of the algorithm, the cost of all paths found by ants as well as the average cost along the simulation were plotted in Figure 5 (for the same Scenario as before). In the beginning ants find random paths, and the plot shows many different costs. However, over time ants tend to converge for paths with more amount of pheromones, i.e., better paths. Then, around cycle 100 all ants tend to converge for the optimal solution: the path with the lowest cost.

Figure 6 shows the simulation results for Scenario 1 with a density of 1.11 $vehicles/km^2$. As expected, as the density of vehicles increase, more paths are found. Moreover, the cost average of paths reduces. This cost reduction can be justified by the

[2]For further details, see: `https://www.nsnam.org/doxygen/classns3_1_1_three_log_distance_propagation_loss_model.html`.

Table 3: Specific configurations for each Scenario. $Distance0$, $Distance1$ and $Distance2$ are, respectively, the beginning of the first (near), second (middle), and third (far) distance fields. Similarly, $Exponent0$, $Exponent1$ and $Exponent2$ are the exponents for the first, second and third fields. Parameters $m0$, $m1$ and $m2$ modulate the fading severity of the model for each field.

	Scenarios			
	1	**3**	**2**	**4**
Vehicles distribution	normal	80/20	normal	80/20
Propagation model	*Three Log Distance*			
$Distance0$ (m)	1		1	
$Exponent0$	2.5		2.5	
$Distance1$ (m)	75		75	
$Exponent1$	5		5	
$Distance2$ (m)	114		114	
$Exponent2$	10		10	
Fading model	*Nakagami-m*			
$m0$	-		1.5	
$Distance1$ (m)	-		60	
$m1$	-		0.75	
$Distance2$ (m)	-		145	
$m2$	-		0	

Table 4: Simulations performed for each scenario.

	Scenario 1				
Vehicles density	0.07	0.14	0.28	0.56	1.11
Path found	-	-	-	yes	yes
	Scenario 2				
Vehicles density	0.07	0.14	0.28	0.56	1.11
Path found	-	-	-	yes	yes
	Scenario 3				
Vehicles density	0.07	0.14	0.28	0.56	1.11
Path found	-	-	-	-	yes
	Scenario 4				
Vehicles density	0.07	0.14	0.28	0.56	1.11
Path found	-	-	-	yes	yes

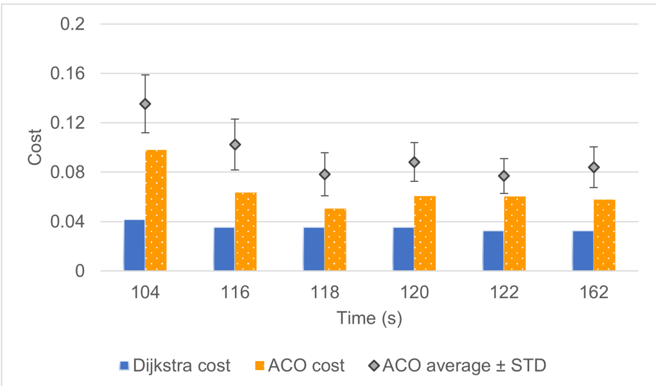

Figure 4: Cost of paths for Scenario 1 with 0.56 $vehicle/km^2$.

increasing number of paths between source and destination. With more path possibilities, the network tends to be less congested, thus reducing the average time to send data packets. Consequently, ants can find paths with the lowest cost.

Figure 7 shows the cost of paths for Scenario 2, with 0.56 $vehicle/km^2$. In such scenario we used the *Nakagami-m* fading model, and more paths were found, when compared with the Scenario 1. According to [36], the use of a deterministic propagation model such as *Three Log Distance* with fading model increases the performance of data transmission because a lower transmission power reduces the signal interference and the packet loss. Although our simulations corroborate with [36], such improvements in the number of paths may suggest wrong transmission power configuration. Therefore, further studies will be necessary in the future works to investigate the effects of different range of power transmission in the simulation results.

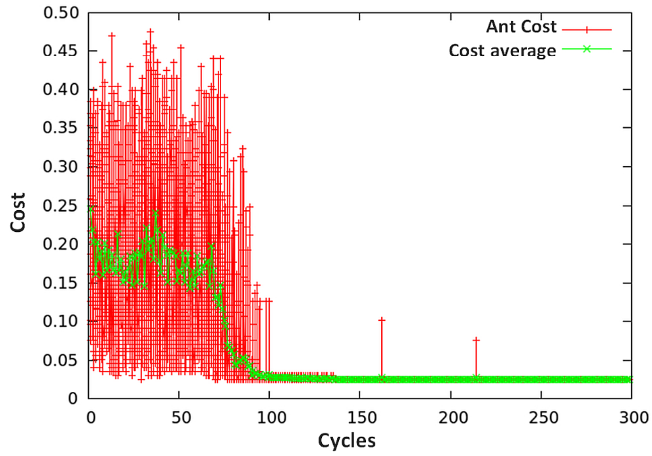

Figure 5: Cost of all paths found by each ant for Scenario 1 with 0.56 $vehicle/km^2$.

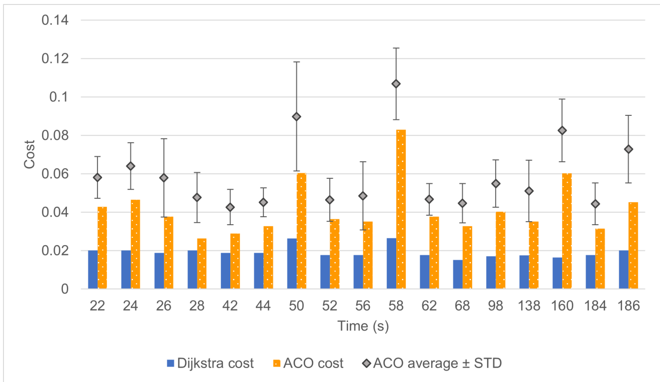

Figure 6: Cost of paths for Scenario 1 with 1.11 $vehicles/km^2$.

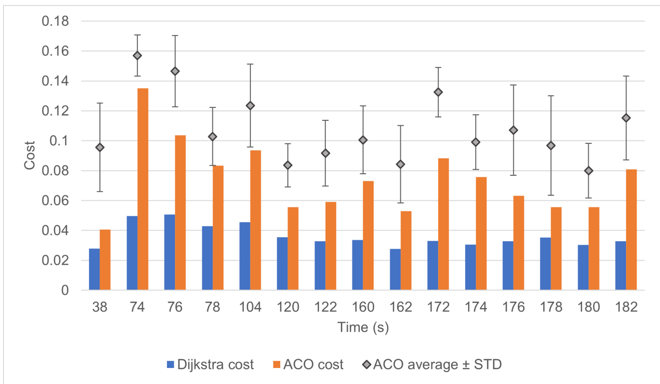

Figure 7: Cost of paths for Scenario 2 with 0.56 $vehicle/km^2$.

In the Scenario 3, 80% of vehicles were distributed in the main avenues while 20% were distributed on the streets. Overall, such network topology leads to less connectivity between vehicles, representing a drawback to the communication. Unlike Scenario 1 where paths were found for the network with 0.56 and 1.11 $vehicles/km^2$, for Scenario 3 paths were found only for the network with 1.11 $vehicles/km^2$ (see Table 4). Such a fact corroborates with the observation of decreased connectivity.

Comparing Scenario 3 and 4 (see Table 4), it is clear what was already seen in Scenario 2 regarding the overall connectivity improvement when using the *Nakagami-m* fading model. Although Scenario 3 uses the *Three Log Distance* propagation model, the use of the fading model in Scenario 4 improved results in a harder situation (with only 0.56 $vehicle/km^2$).

7 Conclusions and Future Work

The world population and the number of motor vehicles on the planet are growing rapidly. This growth brings negative impacts to our environment, such as air and noise pollution, traffic jams and traffic accidents. To deal with such scenario, vehicles need to increase their environment awareness. This can be achieved by enabling vehicles to communicate with their environment within the scope of smart-cities.

Besides V2X communications (V2V and V2I), vehicles could communicate with a large variety of devices. In this work, we considered only *ad hoc* communication between vehicles (VANET). There are many routing algorithms for VANET in the literature. However, benchmarks for comparing such algorithms are still missing. Therefore, we proposed a benchmark for data routing in VANETs.

In our simulations, we observed that several parameters can impact the connectivity in the VANET, for instance: the network density (i.e, the number of vehicles in a given area), the geographical distribution of vehicles, and the transmission power.

The simulation scenarios were configured to better represent the urban traffic in the real world. The propagation and fading models simulate the signal interference caused by the presence of obstacles, like buildings and urban canyons. We observed that such configurations are severe, drastically attenuating the signals transmitted by each vehicle.

In future work we will give some steps ahead by simulating the four scenarios with different levels of power transmission. Such simulation could be useful to define the ideal transmission power range to be used for each scenario. Moreover, in order to better evaluate the ACO algorithm, a new scenario will be created with a time-changing asymmetric density of vehicles.

Acknowledgement

H.S. Lopes thanks to the Brazilian National Research Council (CNPq) for the research grant No. 311785/2019-0, and, also, Fundação Araucária for the research grant PRONEX-042/2018.

Referências

[1] International Organization of Motor Vehicle Manufacturers. “World vehicles in use – all vehicles”. Available in the internet at: `https://www.oica.net/category/vehicles-in-use/`, 2015.

[2] J. Sousanis. “World Vehicle Population Tops 1 Billion Units”. Available in the internet at: `http://wardsauto.com/news-analysis/world-vehicle-population-tops-1-billion-units`, 2016.

[3] International Organization for Standardization. *ISO 21217:2014 Intelligent transport systems – Communications access for land mobiles (CALM) – Architecture*. ISO, Geneve, Switzerland, 2014.

[4] C. K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall, Hoboken, USA, 2001.

[5] H. Sun, C. Wang and B. I. Ahmad, editors. *From Internet of Things to Smart Cities*. CRC Press, Boca Raton, USA, 2018.

[6] S. Kannadhasan, M. Shanmuganantham and R. Nagarajan. “System Model of VANET using Optimization-Based Efficient Routing Algorithm”. In *Proc. International Conference on Advances in Materials Science, Communication and Microelectronics*, Jaipur, India, 2021.

[7] G. K. Marwah and A. Jain. “Congestion-free routing based on a hybrid meta-heuristic algorithm to provide an effective routing protocol by analyzing the significant impacts of QoS parameters in a dynamic VANET environment”. *Journal of Physics: Conference Series*, vol. 2251, no. 012009, 2022.

[8] A. Paul, N. Chilamkurti, A. Daniel and S. Rho. *Intelligent Vehicular Networks and Communications*. Elsevier, Cambridge, USA, 2017.

[9] R. Silva, S. Noguchi, T. Ernst, A. de La Fortelle and W. Godoy Junior. “Standards for Cooperative Intelligent Transportation Systems: a Proof of Concept”. In *The Tenth Advanced International Conference on Telecommunications (AICT)*, pp. 1–6, Paris, France, 2014. International Academy, Research, and Industry Association (IARIA).

[10] R. K. Chauhan and A. Dahiya. “AODV extension using ant colony optimization for scalable routing in VANETs”. *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 2, pp. 241–244, 2012.

[11] B. Dorronsoro, P. Ruiz, G. Danoy, P. Bouvry and L. Tardon. “Towards connectivity improvement in VANETs using bypass links”. In *Proc. IEEE Congress on Evolutionary Computation*, pp. 2201–2208, 2009.

[12] R. R. Sarkar, A. Chakrabarty and M. Z. Rahman. “VANET Routing Protocols in Real-World Mobility Tracing”. In *Proc. IEEE 13th International Conference on Computational Intelligence and Communication Networks*, 2021.

[13] S. Geetha. *VANET: Routing Protocols and Applications*. Lambert Academic Publishing, 2019.

[14] Y. A. Shah, F. Aadil, A. Khalil, M. Assam, I. Abunadi, A. S. Alluhaidan and F. N. Al-Wesabi. “An Evolutionary Algorithm-Based Vehicular Clustering Technique for VANETs”. *IEEE Access*, pp. 14368–14385, 2022.

[15] M. Dorigo and T. Stutzle. *Ant Colony Optimization*. MIT Press, Cambridge, USA, 2004.

[16] R. Silva, H. S. Lopes and W. Godoy Junior. "A Heuristic Algorithm Based on Ant Colony Optimization for Multi-objective Routing in Vehicle Ad Hoc Networks". In *Proc. BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pp. 435–440, 2013.

[17] H. Rana, P. Thulasiraman and R. K. Thulasiram. "MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET". In *Proc. IEEE Congress on Evolutionary Computation*, pp. 2948–2955, 2013.

[18] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. "The self-organizing exploratory pattern of the argentine ant". *Journal of lnsect Behavior*, vol. 3, no. 2, pp. 159–168, 1990.

[19] X. Xiang, Y. Tian, X. Zhang, J. Xiao and Y. Jin. "A Pairwise Proximity Learning-Based Ant Colony Algorithm for Dynamic Vehicle Routing Problems". *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5275–5286, 2022.

[20] M. Perretto and H. S. Lopes. "An ant colony system for large-scale phylogenetic tree reconstruction". *Journal of Intelligent and Fuzzy Systems*, vol. 18, no. 6, pp. 575–583, 2007.

[21] D. Zhao, L. Liu, F. Yu, A. A. Heidari, M. Wang, G. Liang, K. Muhammad and H. Chen. "Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy". *Knowledge-Based Systems*, vol. 216, pp. 106510, 2021.

[22] V. Malik, R. Mittal, J. Singh, V. Rattan and A. Mittal. "Feature Selection Optimization using ACO to Improve the Classification Performance of Web Log Data". In *Proc. IEEE* 8^{th} *International Conference on Signal Processing and Integrated Networks*, 2021.

[23] A. B. Souza, J. Celestino, F. A. Xavier, F. D. Oliveira, A. Patel and M. Latifi. "Stable multicast trees based on ant colony optimization for vehicular ad hoc networks". In *Proc. International Conference on Information Networking (ICOIN)*, pp. 101–106, 2013.

[24] R. Cunha Silva. "Otimização Multiobjetivo Baseada em Colônia de Formigas para o Roteamento e Consumo de Energia em Redes de Sensores sem Fio". Dissertação de mestrado, CPGEI, Universidade Tecnológica Federal do Paraná, Curitiba, Brasil, 2008.

[25] H. T. Friis. "A note on a simple transmission formula". *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[26] M. M. Patil, R. Sadakale, N. Ramesh and R. A. Patil. "An Efficient Bio-inspired Algorithms for Selection of Dynamic Route to Avoid Traffic Flow in VANET". In *Proc.* 12^{th} *International Conference on Computing Communication and Networking Technologies*, pp. 1–5, 2021.

[27] R. Silva. "Proposta de Benchmark para Simulações de Roteamento de Dados em Redes Veiculares *Ad Hoc*". Dissertação de mestrado, CPGEI, Universidade Tecnológica Federal do Paraná, Curitiba, Brasil, 2015.

[28] J. Härri, M. Fiore, F. Fethi and C. Bonnet. "VanetMobiSim: generating realistic mobility patterns for VANETs". In *Proc. of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06)*, pp. 96–97, 2006.

[29] "NS-3: Network Simulator 3". `https://www.nsnam.org/`. Accessed: 06.05.2022.

[30] G. Carneiro, P. Fortuna and M. Ricardo. "FlowMonitor – a network monitoring framework for the Network Simulator 3 (NS-3)". In *Proc. 4th International Conference on Performance Evaluation Methodologies and Tools*. ICST, 2009.

[31] P. L. Frana and T. J. Misa. "An interview with Edsger W. Dijkstra". *Communications of the ACM*, vol. 53, no. 8, pp. 41–47, 2010.

[32] I. Das and J. E. Dennis. "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems". *Structural and Multidisciplinary Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[33] J. Benin, M. Nowatkowski and H. Owen. "Vehicular network simulation propagation loss model parameter standardization in NS-3 and beyond". In *Proc. IEEE Southeastcon*, pp. 1–5, 2012.

[34] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy. "Analytical model for connectivity in vehicular ad hoc networks". *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3341–3356, 2008.

[35] S. Panichpapiboon and W. Pattara-Atikom. "Connectivity requirements for self-organizing traffic information systems". *IEEE Transactions on Vehicular Technology,*, vol. 57, no. 6, pp. 3333–3340, 2008.

[36] M. Stoffers and G. Riley. "Comparing the NS-3 Propagation Models". In *Proc. IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 61–67, 2012.

Annex

A Scenario 1

Table 5: Simulation results of scenario 1.

Vehicle Density (vehicles/km^2)	Simulation instant	Dijkstra cost	ACO cost	ACO vs. Dijkstra (%)	ACO average	STD
0.56	104	0.041727	0.098069	235%	0.135374	0.023477
	116	0.035344	0.063664	180%	0.10237	0.020532
	118	0.035335	0.050468	143%	0.078256	0.017547
	120	0.035342	0.060514	171%	0.08822	0.015657
	122	0.032765	0.060424	184%	0.076837	0.013988
	162	0.03276	0.057949	177%	0.083894	0.016501
1.11	22	0.020092	0.042713	213%	0.058122	0.010853
	24	0.020082	0.046424	231%	0.063936	0.012168
	26	0.018832	0.037694	200%	0.057847	0.020419
	28	0.020087	0.026363	131%	0.047622	0.013003
	42	0.018835	0.028878	153%	0.042624	0.009178
	44	0.018833	0.032642	173%	0.045134	0.007582
	50	0.026388	0.060457	229%	0.089819	0.02839
	52	0.017609	0.036426	207%	0.046392	0.011129
	56	0.0176	0.035157	200%	0.048475	0.017735
	58	0.026402	0.082941	314%	0.106798	0.018576
	62	0.017614	0.037734	214%	0.046682	0.008219
	68	0.015185	0.032649	215%	0.044708	0.010225
	98	0.016987	0.040173	236%	0.054883	0.012312
	138	0.017566	0.035119	200%	0.051053	0.016024
	160	0.016372	0.060278	368%	0.082536	0.016262
	184	0.017612	0.031374	178%	0.044301	0.010843
	186	0.020109	0.045189	225%	0.07276	0.017565

B Scenario 2

Table 6: Simulation results of scenario 2.

Vehicle Density (vehicles/km^2)	Simulation instant	Dijkstra cost	ACO cost	ACO vs. Dijkstra (%)	ACO average	STD
0.56	38	0.027913	0.040567	145%	0.09556	0.029659
	74	0.04973	0.135129	272%	0.156981	0.013669
	76	0.050575	0.103687	205%	0.146491	0.02389
	78	0.04295	0.083279	194%	0.102798	0.019429
	104	0.04552	0.09353	205%	0.123455	0.027729
	120	0.035416	0.055633	157%	0.08356	0.014469
	122	0.03289	0.059185	180%	0.091588	0.021952
	160	0.033529	0.073111	218%	0.100558	0.022688
	162	0.02776	0.052958	191%	0.084231	0.02592
	172	0.032942	0.088278	268%	0.132444	0.016539
	174	0.030599	0.075806	248%	0.09914	0.018282
	176	0.032868	0.063107	192%	0.107062	0.030179
	178	0.035357	0.055556	157%	0.096751	0.033243
	180	0.030341	0.055577	183%	0.079977	0.018278
	182	0.032837	0.080768	246%	0.115209	0.028082
1.11	22	0.02012	0.052787	262%	0.079381	0.020266
	24	0.01888	0.051818	274%	0.072322	0.016083
	26	0.017831	0.037639	211%	0.067254	0.017048
	28	0.018026	0.040157	223%	0.065137	0.017234
	38	0.017578	0.028882	164%	0.047355	0.013126
	40	0.017574	0.038895	221%	0.059993	0.025107
	42	0.018895	0.041504	220%	0.060704	0.017832
	50	0.017594	0.036415	207%	0.058792	0.019179
	52	0.017636	0.049038	278%	0.061386	0.009354
	58	0.016364	0.031393	192%	0.057687	0.016518
	60	0.015181	0.037818	249%	0.072681	0.019295
	62	0.016374	0.031372	192%	0.054303	0.019074
	64	0.026466	0.062793	237%	0.112876	0.035339
	72	0.017656	0.030156	171%	0.052931	0.020225
	94	0.017586	0.03303	188%	0.046803	0.012172
	96	0.017807	0.045294	254%	0.069728	0.015882
	126	0.016335	0.055244	338%	0.080432	0.017345

C Scenario 3

Table 7: Simulation results of scenario 3.

Vehicle Density (vehicles/km^2)	Simulation instant	Dijkstra cost	ACO cost	ACO vs. Dijkstra (%)	ACO average	STD
1.11	70	0.022593	0.06681	296%	0.091346	0.019099
	78	0.01835	0.051923	283%	0.08554	0.02125
	122	0.01637	0.045191	276%	0.068169	0.01504
	136	0.01634	0.035145	215%	0.051345	0.008958
	144	0.023831	0.057713	242%	0.094274	0.02989
	146	0.025125	0.048965	195%	0.087317	0.022718
	150	0.015072	0.040174	267%	0.050266	0.008844
	152	0.016317	0.027612	169%	0.046732	0.011436
	154	0.016325	0.031388	192%	0.052016	0.013993
	156	0.017569	0.028863	164%	0.04638	0.010657
	158	0.016335	0.030106	184%	0.048105	0.011321
	160	0.017603	0.030145	171%	0.048341	0.01308
	162	0.018842	0.026384	140%	0.039431	0.006538
	164	0.017582	0.023882	136%	0.038578	0.008394

D Scenario 4

Table 8: Simulation results of scenario 4.

Vehicle Density (vehicles/km^2)	Simulation instant	Dijkstra cost	ACO cost	ACO vs. Dijkstra (%)	ACO average	STD
0.56	60	0.05049	0.15921	315%	0.185965	0.026284
	64	0.063303	0.171774	271%	0.223349	0.028941
	72	0.034727	0.065545	189%	0.093493	0.025562
1.11	70	0.025144	0.059037	235%	0.093056	0.020607
	88	0.016335	0.031448	193%	0.057551	0.01787
	90	0.018846	0.040172	213%	0.065432	0.022481
	92	0.017587	0.037695	214%	0.065844	0.024846
	100	0.022602	0.077155	341%	0.110554	0.026873
	102	0.017643	0.040227	228%	0.069992	0.018821
	106	0.023926	0.075435	315%	0.10389	0.015547
	110	0.022963	0.078817	343%	0.104738	0.017403
	112	0.013805	0.036382	264%	0.060717	0.016665
	114	0.017575	0.03521	200%	0.056871	0.015029
	116	0.016331	0.033881	207%	0.048796	0.010739
	146	0.017597	0.038932	221%	0.058875	0.015523
	148	0.01632	0.035143	215%	0.047393	0.00632
	150	0.015104	0.025147	166%	0.050542	0.013336
	152	0.01764	0.031412	178%	0.052466	0.013262
	158	0.017581	0.031451	179%	0.043728	0.008729
	160	0.013834	0.02765	200%	0.04134	0.008257
	162	0.015125	0.031398	208%	0.043987	0.010104