A CONTROLLABLE RSU AND VAMPIRE MOTH TO SUPPORT THE CLUSTER STABILITY IN VANET

Ronald Adrian, Selo Sulistyo, I Wayan Mustika and Sahirul Alam

Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia

ABSTRACT

VANET has a dynamic topology that affects cluster formation stability. It influences vehicle's network quality though supporting this stability requires a fast and small cluster formation process. It is necessary because of the rapidly changing condition of the vehicle's position. Moreover, small cluster groups make network quality more evenly distributed among its members. They are essential components in the formation of good clusters in VANET. In the previous research, the algorithm used for this process is based on moth flame optimization. This study proposes modifications in this algorithm to speed up the convergence process, facilitated by a model of turning angles in the moth. Furthermore, the flame at the destination moth is created dynamically to approach the real conditions. The cluster management process uses a fully controlled Road Side Unit on the vehicle's traffic conditions. In the final result, the convergence time is slightly faster, with better results in the throughput at 101.1%, the number of clusters at 58.1%, and the delay at 5.5%, respectively.

KEYWORDS

Algorithm, Cluster, Mfo, Stable & Vanet.

1. INTRODUCTION

Data communication technology is currently experiencing extraordinary developments through the 5G wireless communication systems [1]. Moreover, there is now research on data communication on 6G networks [2]. It encourages the rapid development of other technologies, including wireless vehicle communication supporting smart cars and cities [3]. Widely known as Vehicle Ad-Hoc Network (VANET), it is a component of an intelligent transportation system (ITS), operating on a 5.9 GHz network, which runs in the IEEE 802.11p network standard [4].VANET can make a vital contribution to future vehicles, especially autonomous vehicles [5]. Everything will be controlled using a computer system connected to various sensors through the internet [6]. This internet will be the backbone for exchanging data between vehicles. It is imperative to improve the network quality and stability in VANET [7][8].

There are two types of data communication on VANET. The first one is vehicle-to-vehicle communication (V2V), critical in exchanging data between vehicles in one cluster. It also acts as a Cluster Head (CH) in the connection between vehicles. The second type connects CH to the Road Side Unit (RSU) [9]. The RSU acts like a gateway inside a router, collecting connections from one or several vehicle clusters [10]. It can also communicate between RSUs via wireless or optical networks through a comprehensive deployment [11]. It serves as an access point on the road's side, such as street lighting, traffic light, and cellular tower [12]. The network topology formed on VANET is a mesh network connected, provided it is within the range [13]. Figure 1

shows the connections in VANET under the mesh topology and we use this figure as the simulation topology to make the real environment.



Figure 1. Mesh Topology in VANET

To optimize network connections on VANET, appropriate cluster methods for dynamic nodes have been created [14]. The latest technique uses moth as the core of the algorithm. Moth Flame Optimization (MFO) is a new algorithm in the metaheuristic world that helps stabilize the cluster in VANET, using the Moth Flame Clustering Algorithm for IoV (MFCA-IoV). It involves the movement of a moth in finding a flame [15]. It is a full spiral path, becoming less suitable when fully implemented in VANET because the car trajectory and the signal being issued cannot spiral. Therefore, a modification of the moth algorithm without changing its natural movement is proposed. Parameters are added to adjust the turning angle of the moth in a spiral path, affecting the convergence time in the algorithm. They accelerate the convergence time and slightly straighten the moth movement to match the vehicle on VANET. Static flame behavior is also adjusted. The philosophy of CH in a cluster is approached, where this CH is a moving vehicle, becoming a reference connection for the cluster members. Since it is mobile, parameters are added to make the flame move. Although the movement of CH as a flame is dynamic, modifying this algorithm still guards it. Adjusting the flame does not violate the natural rules in the moth habitat. This is evidenced by a moth species known as the Vampire Moth with targets, such as human blood, as moving entities [16].

In a stable cluster, apart from the formation speed, other aspects such as the network connection should be considered [17]. A good network connection has indicators, including high throughput, whose quality guarantees reliable vehicle communication [18]. The throughput is increased by a cluster creation coefficient combined with vehicle traffic situations. This overloads network management on the RSU, reducing the size of the clusters created. The small cluster size has few vehicle members, thereby increasing throughput and connections to very few people's access points. This coefficient is also beneficial in managing work scheduling from RSU. Several RSUs

are turned on in a low traffic condition, with only a few accommodating connections to CH. However, all RSUs can be maximized in a high traffic condition. This coefficient dynamically adjusts based on the traffic conditions. Finally, three methods are proposed to stabilize the cluster, compared to the previous research. These are the turning angle setting on the moth, flame movement, and cluster size coefficient that might change dynamically. The results obtained are validated with the previous method to show the research position.

The remaining sections of this research are structured as follows: Section II talks about the related work that support our research. Section III describes the proposed method. Section IV shows the results, while Section V gives the brief conclusion.

2. RELATED WORK

Studies show that cluster techniques in VANET have been explored [14]. It initially began with node proximity to the centroid as in K-Means with a central algorithm to further develop cluster methods. Modifications of K-Means slightly shift the direction to K-Medoids, whose algorithm has advantages on the accuracy required by VANET. Both are well-known algorithms used for cluster processes in various fields. However, their computation process is slow, with a lot of data clustering. This underlies the research on clusters in VANET using various optimization algorithms. It has switched to optimization methods to help determine optimal, stable, fast results and accommodate dynamic nodes, using a bio-inspired algorithm. Apart from the clustering process, this algorithm is also used in the routing process on VANET [19].

A bio-inspired algorithm is based on the behavior of animals, plants, or other natural phenomena. These algorithms do vary, though those used in VANET are centralized to several algorithms [19]. Several members of bio-inspired in current research are Particle Swarm Optimization (PSO), Gray Wolf Optimization (GWO), Ant Colony Optimization (ACO), and Genetic Algorithm (GA) [20]. There are developments in the algorithm-based method to find the best solution. The development of bio-inspired algorithms on problems in VANET spawned new terms. PSO was developed into a Clustering (CLPSO) and Multi-Objective PSO (MOPSO) algorithm, creating clusters on VANET [21].

CLPSO is used in Mobile Ad-Hoc Network (MANET) clustering [22]. It has a static topology that can be applied to VANET with dynamic topology by computing one snapshot of the process. This feature makes the VANET topology change in a certain time unit. Most algorithms in MANET can be applied in VANET, although it is not optimal [23]. CLPSO uses a weight to perform the clustering scheme in mobile ad-hoc networks to find the optimal number of clusters efficiently supported by the network. Matrix calculations, including transmission power, mobility, and mobile node battery power, are involved. A weight is respectively assigned to all parameters. The information and members of the cluster heads are stored in each PSO particle. The results have been compared with Weighted Clustering Algorithm (WCA) and Divided Range Particle Swarm Optimization (DRPSO), giving a better technique than two previous algorithms. In Multi-Objective PSO (MOPSO), inter-cluster and intra-cluster traffic management is carried out by the cluster head. It uses several measurement matrices, including node level, transmission power, and mobile node battery power consumption. This method simultaneously provides a series of solutions using the optimal front Pareto. The results obtained were compared with WCA and CLPSO, and tests carried out using different metrics [24].

The next research related to the bio-inspired algorithm in VANET is an Efficient Optimization Technique for Node Clustering in VANETs Using Gray Wolf Optimization (ICGWO). This study complements the methods developed in cluster formation. The vehicle cluster functions as a means of data communication. Cluster Heads (CHs) selected to manage all its members,

helping maintain communication between clusters in a system on VANET. A stable cluster has a longer life, improving network performance. The algorithm proposed aims to provide optimal solutions, useful in facilitating and strengthening data communication. The parameters used in ICGWO include grid size, load balance factor (LBF), transmission range, speed, and vehicle direction. The results obtained were compared with previous research, specifically Multi-Objective Swarm Optimization (MOPSO) and Comprehensive Learning Particle Swarm Optimization (CLPSO). During testing, they vary the critical parameters in ICGWO to determine the effectiveness of the proposed algorithm. The key parameters are grid sizes, several vehicles, and transmission ranges. For the algorithm's effectiveness, ICGWO chooses 10% of the nodes as CHs, while CLPSO and MOPSO take 13% and 14%, respectively [25].

Ant Colony Optimization (ACO) is an old algorithm with a new modification in VANET, identified through Ant Colony Optimization (ACO) Based Clustering Algorithm for VANET (CACONET). It aims to improve communication efficiency, using several techniques, such as vehicle node grouping, data aggregation, and message delivery. In this study, CACONET formed an optimal cluster to support strong communication. The results of this study were compared with MOPSO and CLPSO. The model also uses the same variation as ICWGO by varying the number of vehicles, the network grid's size, and the range of transmission nodes. Optimization is conducted by considering the transmission range, direction, and speed of the vehicle. CACONET shows better results than MOPSO and CLPSO [26], Other researchers resumed research on clustering based on bio-inspired algorithms. It was titled the novel Clustering Algorithm, centred on Fire Moth Optimization for VANET (CAMONET). It is based on clustering using the moth flame optimization (MFO) algorithm to produce clusters with strong transmissions. This method is evaluated experimentally using well-known techniques, such as the optimization of multiobject particle crowds, the optimization of a comprehensive crowd of learning particles, and the grouping algorithm based on the optimization of ant colonies for VANET. The results are obtained by modifying the number of vehicles, network grid size, and transmission range of the node. Other important factors considered are speed, direction, and transmission range. The final results show that CAMONET has an optimal cluster formation efficiency compared to previous studies, CLPSO, MOPSO, and CACONET [27]. This research introduces and paves the way for further exploration of the moth-based algorithm.

Moth Flame Optimization (MFO) is a bio-inspired algorithm inspired by the moth behavior of finding the flame. Similar to the butterfly, the moth keeps its lifetime through metamorphosis cycles, as seen in its transformation from the larva to an adult insect [28]. The most interesting fact from a moth is a special genus, *Calyptra* [16]. For instance, *Calyptra Thalictri* can pierce and feed on human blood. Known as a vampire moth, it is the main inspiration in resolving the VANET problems in this paper [29].

The Moth Flame Clustering Algorithm for the Internet of Vehicle (MFCA-IoV) is the current VANET clustering method, adopting the moth movement [15]. A static flame is used in the MFCA-IoV, even though the vehicle movement in VANET is dynamic. The moth can represent dynamic vehicle movements. The flame moves dynamically suppose a cluster is formed on a mobile infrastructure with CH as the center of the connection. This algorithm's weakness is that there is no more in-depth exploration regarding the philosophy of the moth movement on VANET. It is a research gap that can be explored further so that the network performance on the vANET can be maximized. To accommodate this flame movement, further adjustments to the existing MFO algorithm are needed. This dynamic VANET infrastructure can be accommodated with few modifications to the main equation in the MFO algorithm.

3. PROPOSED METHOD

The original moth algorithm is known as Moth Flame Optimization (MFO). It has come with the assumption that the candidate solutions are moth. The problem variables are the moth position in space. In Equation 1, moth movements can be written in a matrix, representing their advantages compared to other heuristic algorithms. It is caused by moth movement in 1D, 2D, 3D, or hyper-dimensional fields,

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & m_{2,3} & \cdots & m_{2,d} \\ m_{3,1} & m_{3,2} & m_{3,3} & \cdots & m_{3,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & m_{n,3} & \cdots & m_{n,d} \end{bmatrix}$$
(1)

where n is the number of moths and d is the number of variables (dimension).

The moth position referring to a fitness value is stored in an array, as shown in the structure in Equation 2,

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ OM_3 \\ \vdots \\ OM_n \end{bmatrix}$$
(2)

where *n* is the number of moths.

The flame has the same array structure as the moth, shown in Equation 3.

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & F_{1,3} & \cdots & F_{1,d} \\ F_{2,1} & F_{2,2} & F_{2,3} & \cdots & F_{2,d} \\ F_{3,1} & F_{3,2} & F_{3,3} & \cdots & F_{3,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{n,1} & F_{n,2} & F_{n,3} & \cdots & F_{n,d} \end{bmatrix}$$
(3)

The flame can be in a multi-dimensional position like the moth. It must be in the same dimension in the 2D area. From Equation 4, the flame storage model's corresponding fitness values are the same as the moth storage model.

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ OF_3 \\ \vdots \\ OF_n \end{bmatrix}$$
(4)

Moth movement is limited by upper and lower bounds to minimize the flame's searching area, accelerating convergence. It is presented in Equation 5 for upper bounds and Equation 6 for lower bounds, respectively.

$$UB = [UB_1, UB_2, ..., UB_{n-1}, UB_n]$$
(5)

$$LB = [LB_1, LB_2, \dots, LB_{n-1}, LB_n]$$
(6)

83

The spiral movement is used as the main process in the MFO algorithm. The mathematical model of the moth movement to flame is shown in Equation 7,

$$S_i(M_i, F_i) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_i$$
(7)

where D_i represents the distance of the *i*-th moth for the *j*-th flame, *b* is a constant for defining the shape of the logarithmic spiral, and t is a random number in [1, 1],

$$D = \left| F_i - M_i \right| \tag{8}$$

where M_i indicates the *i*-th moth, F_j the *j*-th flame, and D_i the distance of the *i*-th moth for the *j*-th flame. The moth movement is depicted in Figure 3.



Figure 2. Spiral path of moth movement

The number of flames is reduced with an increase in iteration to overcome the problem when the moth searches in different locations. The movement patterns reduce the value of the best solutions obtained previously. Equation 9 shows how to reduce the amount of flame gradually.

$$Flame number = round\left(N - 1 * \frac{N - 1}{T}\right)$$
(9)

In each iteration, the computational load of the MFO algorithm depends on the number of moths and variables, the maximum number of iterations, and the flame sorting mechanism. The computation gradually slows down with time. Therefore, a quick sorting method is needed to ease the MFO computational workload. The work of quicksort is modelled in Equations 10 and 11.

$$O(MFO) = O(t(O(Quick \ sort) + O(position \ update)))$$
(10)

$$O(MFO) = O(t(n^2 + n \times d)) = O(tn^2 + tnd)$$
(11)

The formation process using the modified MFO algorithm is shown in Figure 3. The cluster formation stages begin with data collection on the vehicle's position, direction, and speed. The topology created in this simulation is mesh type, allowing all vehicles to communicate directly with other nearby vehicles. However, this mesh topology is limited to only one vehicle cluster whose central gateway is on each cluster. The next step is calculating the distance between

vehicles to determine the entry and exit of cluster members. Because of the dynamic nature of the topology in VANET, changes in cluster membership are likely to occur. Therefore, their maintenance prioritizes vehicles with the closest distance to CH, adopting the K-Means algorithm's clustering process.

Finding a flame as a representation of the MFO algorithm is shown in step 4, where the search space is initialized from the moth position. Virtually, the moth moves to find the best solution in each iteration. The solutions obtained at each iteration are evaluated, and the best is maintained, determining the number of clusters formed in the VANET network. This condition is checked every second to maintain its stability.



Figure 3. Scheme of clustering process in CR-VMV.

Vehicle movement simulations were conducted using SUMO software. Data from SUMO is in the form of XML, further processed using the CR-VMV algorithm. The data contents include the position, speed, acceleration, direction, and type of vehicle. The vehicle position in each data captured is symbolized by the x and y coordinates, as shown in Equation 12 where vp is a vehicle position. It is the direction of vehicles through 90° is going the west and 270° is going to the east.

$$vp = \begin{bmatrix} x, y \end{bmatrix} \tag{12}$$

$$vd = \begin{cases} 50\\270^{\circ} \end{cases}$$
(13)

The vehicle direction is symbolized in Equation 13, used as vehicle identification in a cluster. The nearby vehicles in a different direction are excluded from the cluster because they are included in the minority. They can damage the stability of the system formed when putting in a cluster with a homogeneous movement direction. To distinguish the direction of motion of the vehicle, the degree of wind is used. The north is symbolized by zero degrees, the east direction is 90 degrees, while the west direction is 270 degrees, as presented in Equation 14 where vs is a vehicle speed and ms is a maximum speed (kph) that can be achieved by a vehicle.

$$vs = \begin{cases} 0 \ kph \\ ms \end{cases}$$
(14)

Another critical parameter is the vehicle speed. Uniform speed in the adjacent area makes the vehicles join in the same cluster. It also considers the motion direction explained previously. Existing systems on the VANET infrastructure cannot regulate vehicle speed. This depends on the behavior of the driver. The system only takes speed data from each vehicle, processed as a component of cluster formation. The speed is also very diverse, based on the specifications of various types of vehicles. For example, the speed and acceleration of a bus are different from a city car. Essentially, city cars may have a higher acceleration even though the top speed is less than the bus. Psychologically, the level of acceleration will be driving. In this simulation, four types of vehicles are limited, including cars, SUVs, buses, and trucks. The acceleration and speed are determined according to the average specifications of several well-known vehicle manufacturers.

In the SUMO simulation, vehicle traffic is gradually modelled from quiet to dense. The maximum number of vehicles is set at 5000 per hour with a breakdown of 2000 cars, 1500 SUVs, 1,000 buses, and 500 trucks per hour. This traffic flow is cumulative in two directions, with 2500 vehicles per hour in each direction. The length of the road used is a 1000 meters' highway, chosen to ease vehicles' observation when the algorithm was running. More details on setting parameters are shown in Table 1.

Parameter	Value
Type of Road	Highway
Length of Road	1000 m
Number of Lanes	6
Vehicle Traffic Flow	5000 vehicles per hour
Types of Vehicle	Car, SUV, Bus, and Truck
Simulation Time	100 s
Transmission Range	300 m
Vehicle Movement	Random
Maximum Lanes Speed	120 kph
Gap between Vehicles	1.5 m

Table 1. Simulation Parameters

Parameters were added to the MFO equation's core to the moth's turning angle and the flame movements. It can accelerate or deaccelerate the spiral movement of the flame. This can be modeled mathematically as shown in Equation 15,

$$S_i(M_i, F_j) = D_i \cdot e^{qbt} \cdot \cos(2\pi t) + (F_j \cdot e^t)$$
(15)

where q is turning value with the range of 0 to 1, and e^t is a component that becomes a dynamic position.

Equation 16 is the flame movement (F) that consists of the speed (v) and time (t) components. This causes F to move within a certain time.

$$F = v.t \tag{16}$$

Virtually, the two equations are formed to accommodate the vehicle movement in a 2dimensional plane. Sine and cosine waves are movements that should be matched with the environment in VANET. To apply this, the sinewave can be neutralized by the cosine wave, and the two entities are included in the core equation of the CR-VMV. The running equation in the simulation is Equations 17 and 18.

$$S_i(M_i, F_i) = D_i \cdot e^{qbt} \left(\cos(2\pi t) \cdot X - \sin(2\pi t) \cdot Y \right) + (F_i \cdot e^t)$$
(17)

$$S_{i}(M_{i}, F_{i}) = D_{i} \cdot e^{qbt} \left(sin(2\pi t) \cdot X - cos(2\pi t) \cdot Y \right) + (F_{i} \cdot e^{t})$$
(18)

The cluster size coefficient (*CSC*) is also introduced. This is based on the mobile RSU further developed into a new coefficient to control the RSU. The modeling is shown in Equation 19, where *CSC* is a cluster size coefficient and *nov* is the number of vehicles.

$$CSC = \frac{max(nov)}{mean(nov)}$$
(19)

Conventional technology for RSU turns them on regardless of traffic conditions. A fully controlled RSU adapts to road conditions. In case the road is busy, all of them can be used. However, only a few may suppose the road is deserted. This reduces the number of clusters, increasing the throughput between CH and RSU. There is a surge in the number of cluster members in this condition, anticipated by creating a network relay concept to divide the network underneath. It is a network hierarchy consisting of cores, distribution, and access. The backbone is the core network in VANET. The RSU represents the distribution network, while CH handles the access network.

The proposed method to get useful and optimal results in strengthening network performance is evaluated through several trials. The following pseudo-code is used in making this simulation program.

Algorithm CR-VMV

- 1: Input: position, direction, and speed.
- 2: Generate a mesh among the vehicles.
- 3: Calculate the distance among the vehicles.
- 4: Initialize a search area from the Moths.
- 5: Moth_Pos = zeros(No_Of_Moth, No_Of_Variable);
- 6: **for** i=1:No_Of_Moth
 - **for** j=1:No_Of_Variable
- 7: Moth_Pos(i,j,1) = ((Upper_Bound(1) Lower_Bound(1)) * rand()) + Lower_Bound(1); Moth_Pos(i,j,2) = ((Upper_Bound(2) - Lower_Bound(2)) * rand()) + Lower_Bound(2);
- 8: end

9: **end**

10: **while** (Iteration == Max_Iteration

International Journal of Computer Networks & Communications (IJCNC) Vol.13, No.3, May 2021						
	or the solution equal to the stall iteration==100)					
11:	for i=1:No_Of_Moth					
12:	Find the Fitness_Of_Moth					
13:	Fitness_Of_Moth = Cost_Function(Moth_pos);					
14:	Mothi_Of_Clusters == 0					
15:	Mothi_Cluster_Fitnes == NaN					
16:	No_Of_Node_Selected = 0 (to create a cluster, make all the nodes available)					
17:	Member_Status = false, Solution_Status = false					
18:	if Solution_Status == true					
19:	Cluster_Fitnes = Cluster_Fitnes + Cost_Function(ED(clusterMember));					
20:	Cluster_Fitnes < BestSolCluster_FitnesIteration, BestSol = Mothi;					
21:	end					
22:	end					
23:	[Fitness_sortedFitness_sorted_I] = sort(FitnessOfMoth);					
24:	sorted_population = Moth_Pos(Fitness_sorted_I);					
25:	Update the position_best_flame obtained so far					
	a) BestFlameScore = Fitness_sorted(1);					
	b) BestFlamePos = sorted_population(1, :, :);					
26:	Update the position of moths according to flame.					
27:	for 1=1:No_Of_Moth					
28:	for j=1:No_Of_Variable					
	a) calculate distance to the Flame (1)					
	b) calculate Moth_Pos using CR-VMV					
20.	c) update the position of moth.					
29:	enu					
30: 31:	Convergence, curve(iteration) = BestElemeScore:					
51.	$convergence_curve(neration) - Destructionescore,$					
	stanteration – stanteration + 1,					
37.	stall teration -0 :					
32.	and					
33. 34.	iteration $-$ iteration $+1$:					
3^{-1} . Instanton – Instanton + 1, 35. end						
36: Output: TottalnoOfCluster = BestSol(iteration)/con:						
*ED = Euclidian Distance						

4. SIMULATION RESULTS

The proposed algorithm is validated using unimodal and multi-modal benchmarking methods. This was also carried out in previous studies that proposed a new algorithm or a modification of an existing algorithm [30]. The equation used to carry out this test is by the standards used in the original MFO study [28]. The mathematical modelling used in this work is shown in Table 2.

Function	Dim	Range	Shift Position	Fmin
$f_1(x) = \sum_{i=1}^n x_i^2$	100	[-100, 100]	[-30,-30,,-30]	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	100	[-10, 10]	[-3,-3,,-3]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	100	[-100, 100]	[-30,-30,,-30]	0
$f_4(x) = \sum_{i=1}^{n} (x_i^4 + random[0,1])$	100	[-1.28, 1.28]	[-0.25,,-0.25]	0
$f_5(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	100	[-5.12, 5.12]	[-2,-2,,-2]	0

International Journal of Computer Networks & Communications (IJCNC) Vol.13, No.3, May 2021 Table 2. Benchmark Multimodal and Unimodal

Table 3. Convergence Curve of Benchmark Multimodal and Unimodal Test Function





International Journal of Computer Networks & Communications (IJCNC) Vol.13, No.3, May 2021

The results from the benchmark method are shown in Table 3, including two tests. The unimodal test is represented by F1-F3, while F4-F5 represents multi-modal. The search dimensions for the algorithm testing are the same for all test functions, though the range differs. Based on the technical report introduced by previous research, each function has its naming [31]. F1 is a Simple Sphere Function, F2 is the Shifted Sphere Function, F3 is the Shifted Schwefel's Function, F4 is the Schaffer's Function, and F5 is the Shifted Spacing Function. Each of these functions is different. Hopefully, this can be a solution for evaluating a new formula to be used in a system. When proven to be good, the results continue to the next process. The tests are conducted with five functions (F1-F5). The results obtained by CR-VMV are superior to the previous method. The best convergence value obtained by CR-VMV is the provision for the proposed method in VANET. The proposed turning angle symbolized by q in the Moth core equation accelerates the convergence speed. If q approaches zero, the angle enlarges and approaches a straight line, accelerating the moth to the flame. The detailed results of this test are shown in Table 3.

Figure 4 is the end-to-end delay resulting from the simulation. The delay generated by CR-VMV is not inferior to MFCA-IoV. The delay generated by the previous research was adopted for the CR-VMV system since it was good enough. From a closer look, the method is somewhat superior at low density, and this may result from the modification of the movement pattern of the moth approaching a straight line when heading towards the flame. This pattern is good at low density compared to high density. High density makes the method inappropriate in the delay production because the computational process has a heavy burden in high density, disturbing the delay. On average, the proposed method has a delay of 5.5% compared to the previous method. For this reason, it contributes to delay optimization.



Figure 4. End to End Delay Comparison

Figure 5 shows the impact of using the cluster size coefficient (CSC). The number of clusters can be reduced to a minimum while still considering the quality of the connection. Technically, the reduction increases cluster size, lowering the number of CH connected to RSUs. Furthermore, it increases the throughput compared to the previous method. The reduced RSU burden lowers efficient RSU deployment costs since the equipment installed in its system does not require high technology, reducing the price. The number of RSUs to be deployed along the road multiplied by the price per unit receives better savings. In this case, the solution offered accommodates two throughput and economics fields, though an in-depth review is needed in terms of economic claims. The decrease in the number of clusters from various types of road density conditions was 58.1%.



Figure 5. Cluster Size Comparison



Figure 6. Throughput Comparison

Another effect of decreasing the number of clusters is a throughput increase compared to the previous method. The resulting throughput is better than the previous method, as shown in Figure 6, which increases with increasing vehicle density. Based on the average increase of 101.1%, CR-VMV is far better than the previous method due to a significant decrease in the number of clusters. The measurements for this throughput are made between the cluster head and the RSU.

Reducing the number of clusters is also disadvantageous since it leads to a higher burden on the cluster head, which requires an excellent device to provide a network to all its members. In the real world, very diverse vehicles can be seen, though most of them are the latest release. Therefore, vehicle priority is a gateway to the cluster head. Expensive vehicles are identical with good specifications. They are equipped with good network technology and more expensive buses, with higher specifications than city cars. Physically, the bus might also act as a cluster head because it has a higher dimension, a better electrical system in terms of power support, and a slower and more stable movement pattern. The same conditions apply to trucks, which are also the main vehicles with high priority as cluster heads. In case the network load by the cluster head is too high, the unmerging method can be used as in the ABNC method in previous studies [9].

5. CONCLUSION AND FUTURE WORK

To improve cluster stability, the CR-VMV method is proposed. It consists of several parts, including the modification of the MFO algorithm used in previous studies. It also contains the addition of the coefficient feature to reduce the number of clusters formed in the MFO algorithmbased system. We modify the MFO algorithm with a new approach, namely the vampire moth. The search for prey that moves on the vampire moth has the same analogy as the search for cluster head on VANET. The goal is to maximize all existing resources on the VANET so that network performance can increase. We also add a controller to the RSU for power savings. It is important to support green energy in the future.

Convergence time in our proposed method is better than the previous study, as tested using the benchmarking model. The throughput has been compared in the other results, where the number of clusters and end-to-end delay (E2ED) increases at 101.1% and decreases at 58.1% and 5.5%, respectively. However, there are shortcomings in terms of increasing the delay parameter. In the

proposed method, there is only a slight reduction in the delay value. This might be achieved by the exploration of suitable algorithms in the future. Economic analysis of the use of RSUs through the CR-VMV also needs to be conducted to establish certain economic-based efficiency. The method proposed can open a research area in smart RSU technology as a more reliable network provider.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGEMENTS

The research is funded by the RTA program in 2020 from the Research Directorate of Universitas Gadjah Mada.

REFERENCES

- X. Duan, Y. Liu, and X. Wang, "SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 120–127, 2017, DOI: 10.1109/MCOM.2017.1601160.
- [2] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," *IEEE Open J. Commun. Soc.*, vol. 1, no. July, pp. 957–975, 2020, DOI: 10.1109/OJCOMS.2020.3010270.
- [3] P. Santi, "Vehicular Networks," in *Mobility Models for Next Generation Wireless Networks*, Wiley Telecom, 2012, pp. 139–152.
- [4] Y. Harkat and A. Amrouche, "Vehicle density, vehicle speed, and packet inter-arrival time analysis in IEEE 802.11p EDCA based VANETs," in 2018 International Conference on Signal, Image, Vision and their Applications (SIVA), Nov. 2018, pp. 1–6, DOI: 10.1109/SIVA.2018.8661070.
- [5] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019, DOI: 10.1109/COMST.2018.2869360.
- [6] C. Liu, K. T. Chau, D. Wu, and S. Gao, "Opportunities and Challenges of Vehicle-to-Home, Vehicle-to-Vehicle, and Vehicle-to-Grid Technologies," *Proc. IEEE*, vol. 101, no. 11, pp. 2409–2427, Nov. 2013, DOI: 10.1109/JPROC.2013.2271951.
- [7] N. M. Al-Kharasani, Z. A. Zukarnain, S. K. Subramaniam, and Z. M. Hanapi, "An Adaptive Relay Selection Scheme for Enhancing Network Stability in VANETs," *IEEE Access*, vol. 8, pp. 128757– 128765, 2020, DOI: 10.1109/ACCESS.2020.2974105.
- [8] S. Sulistyo, S. Alam, and R. Adrian, "Coalitional Game Theoretical Approach for VANET Clustering to Improve SNR," J. Comput. Networks Commun., vol. 2019, pp. 1–13, Jul. 2019, DOI: 10.1155/2019/4573619.
- [9] R. Adrian, S. Sulistyo, I. Mustika, and S. Alam, "ABNC: Adaptive Border Node Clustering Using Genes Fusion Based on Genetic Algorithm to Support the Stability of Cluster in VANET," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 1, pp. 354–363, Feb. 2020, DOI: 10.22266/ijies2020.0229.33.
- [10] M. Alawi, E. Sundararajan, R. Alsaqour, and M. Ismail, "Gateway selection techniques in aheterogeneous vehicular network: Review and challenges," *Proc. 2017 6th Int. Conf. Electr. Eng. Informatics Sustain. Soc. Through Digit. Innov. ICEEI 2017*, vol. 2017-Novem, pp. 1–6, Nov. 2018, DOI: 10.1109/ICEEI.2017.8312425.
- [11] D. Kim, Y. Velasco, W. Wang, R. Uma, R. Hussain, and S. Lee, "A New Comprehensive RSU Installation Strategy for Cost-Efficient VANET Deployment," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 1–1, 2016, DOI: 10.1109/TVT.2016.2598253.
- [12] M. Patra and C. S. R. Murthy, "Performance Evaluation of Joint Placement and Sleep Scheduling of Grid-Connected Solar Powered Road Side Units in Vehicular Networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1197–1209, Dec. 2018, doi: 10.1109/TGCN.2018.2864152.
- [13] Sung-Gi Min, Seong-Mun Kim, and Jinwoo Nam, "Extended wireless mesh network for VANET with a geographical routing protocol," in *11th International Conference on Wireless*

Communications, Networking and Mobile Computing (WiCOM 2015), 2015, pp. 6 .-6 ., DOI: 10.1049/cp.2015.0746.

- [14] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A Comparative Survey of VANET Clustering Techniques," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 657–681, 2017, DOI: 10.1109/COMST.2016.2611524.
- [15] M. F. Khan, F. Aadil, M. Maqsood, S. H. R. Bukhari, M. Hussain, and Y. Nam, "Moth Flame Clustering Algorithm for the Internet of Vehicle (MFCA-IoV)," *IEEE Access*, vol. 7, pp. 11613– 11629, 2019, DOI: 10.1109/ACCESS.2018.2886420.
- [16] J. M. Zaspel, C. H. Scott, S. R. Hill, R. Ignell, V. S. Kononenko, and S. J. Weller, "Geographic Distribution, Phylogeny, and Genetic Diversity of the Fruit- and Blood-Feeding Moth Calyptra thalictrum Borkhausen (Insecta: Lepidoptera: Erebidae)," J. Parasitol., vol. 100, no. 5, p. 583, Oct. 2014, doi: 10.1645/13-484.1.
- [17] R. Hajlaoui, E. Alsolami, T. Moulahi, and H. Guyennet, "Construction of a stable vehicular ad hoc network based on hybrid genetic algorithm," *Telecommun. Syst.*, vol. 71, no. 3, pp. 433–445, Jul. 2019, DOI: 10.1007/s11235-018-0513-6.
- [18] A. A. Mu'azu, L. T. Jung, I. A. Lawal, and P. A. Shah, "Throughput Measurement for the Guaranteed QoS Real-time Traffic Flows in VANETs," *Proceedia - Soc. Behav. Sci.*, vol. 129, no. Supplement C, pp. 297–304, May 2014, DOI: 10.1016/j.sbspro.2014.03.680.
- [19] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 2, pp. 843–867, 2015, DOI: 10.1109/COMST.2014.2371828.
- [20] R. Adrian, S. Sulistyo, and I. W. Mustika, "A Study on Communication System in VANET," in 2018 4th International Conference on Science and Technology (ICST), Aug. 2018, pp. 1–6, DOI: 10.1109/ICSTC.2018.8528640.
- [21] G. Husnain, S. Anwar, and F. Shahzad, "Performance evaluation of CLPSO and MOPSO routing algorithms for optimized clustering in Vehicular Ad hoc Networks," in *Proceedings of 2017 14th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2017*, Jan. 2017, pp. 772–778, DOI: 10.1109/IBCAST.2017.7868141.
- [22] W. Shahzad, F. A. Khan, and A. B. Siddiqui, "Clustering in Mobile Ad Hoc Networks Using Comprehensive Learning Particle Swarm Optimization (CLPSO)," in *Communications in Computer* and Information Science, vol. 56, Springer, Berlin, Heidelberg, 2009, pp. 342–349.
- [23] R. S. Hande and A. Muddana, "Comprehensive survey on clustering-based efficient data dissemination algorithms for VANET," in *International Conference on Signal Processing*, *Communication, Power and Embedded System, SCOPES 2016 - Proceedings*, Oct. 2017, pp. 629– 632, DOI: 10.1109/SCOPES.2016.7955516.
- [24] H. Ali, W. Shahzad, and A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 12, pp. 1913–1928, 2012, DOI: 10.1016/j.asoc.2011.05.036.
- [25] M. F. Khan, F. Aadil, M. Maqsood, S. Khan, and B. H. Bukhari, "An efficient optimization technique for node clustering in VANETs using gray wolf optimization," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 9, pp. 4228–4247, 2018, DOI: 10.3837/tiis.2018.09.007.
- [26] F. Aadil, K. Bashir Bajwa, S. Khan, N. M. Chaudary, and A. Akram, "CACONET: Ant Colony Optimization (ACO) Based Clustering Algorithm for VANET," 2016, DOI: 10.1371/journal.pone.0154080.
- [27] Y. A. Shah, H. A. Habib, F. Aadil, M. F. Khan, M. Maqsood, and T. Nawaz, "CAMONET: Moth-Flame Optimization (MFO) Based Clustering Algorithm for VANETs," *IEEE Access*, vol. 6, pp. 48611–48624, 2018, DOI: 10.1109/ACCESS.2018.2868118.
- [28] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Syst.*, vol. 89, pp. 228–249, 2015, DOI: 10.1016/j.knosys.2015.07.006.
- [29] J. M. Zaspel, R. Zahiri, C. Food, I. Agency, D. Janzen, and M. Hoy, "A preliminary phylogeny of the vampire moths and their fruit-piercing relatives based on molecular data (Noctuidae: Calpinae)," no. December 2010.
- [30] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," Int. J. Comput. Math., vol. 77, no. 4, pp. 481–506, Jan. 2001, DOI: 10.1080/00207160108805080.
- [31] P. N. Suganthan *et al.*, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," 2005. Accessed: Jul. 05, 2020. [Online]. Available: http://www.cs.colostate.edu/~genitor/functions.html.

AUTHORS

Ronald Adrian received the Bachelor's and Master's degrees in Electrical Engineering from Universitas Gadjah Mada, Indonesia, in 2013 and 2015. He is currently a Ph.D. student at Electrical Engineering at Universitas Gadjah Mada, supported by the Indonesia Endowment Fund for Education (LPDP). He has been a lecturer at Universitas Gadjah Mada since 2016. His research interest is wireless communication, especially in an adhoc network

Prof. Selo Sulistyo is currently a Full-Time Professor of Information and Communication Technology in the Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada. He received his M.Sc. and a Ph.D. degree from Agder University, Norway, in 2003 and 2012, respectively. His current research interests include vehicular communications, software modeling, IoT, network security, and connected objects.

Dr. I Wayan Mustika is an Assistant Professor who finished a Bachelor's Degree in Electrical Engineering from Universitas Gadjah Mada, Indonesia, in 2005 and a Master's degree in computer engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2008. He got a Ph.D. degree in informatics from Kyoto University in 2011, with the Japan International Cooperation Agency (JICA) scholarship. His current research interests include heterogeneous networks, radio resource management in cellular networks, and IoT. He was an awardee of the IEEE

Award in 2011. He is also a member of the IEEE.

Sahirul Alam received a Bachelor's degree in 2013 from Universitas Brawijaya, Indonesia, and a Master's degree from Universitas Gadjah Mada in 2017, respectively, in Electrical Engineering, Indonesia. He is currently pursuing a Ph.D. degree in the intelligent transportation system at Universitas Gadjah Mada since Aug 2018. His research interests include vehicular networking, fuzzy systems, optimization algorithms, radio resource management in wireless networks, and spectrum sharing.

VTS Japan 2010 Young Researcher's Encouragement Award and the IEEE Kansai Section Student Paper



