A new energy-aware method for load balance managing in the fog-based vehicular ad hoc networks (VANET) using a hybrid optimization algorithm

Ren Qun¹ | Seyedeh Maryam Arefzadeh²

¹ Department of Electronic and Information Engineering, Bozhou University, Bozhou, Anhui 236800, China
² Department of Control Engineering, Bushehr Branch, Islamic Azad University, Bushehr, Iran

Abstract
Fog-based VANETs (Vehicular Ad hoc NETworks) is a new model with vehicular cloud and fog computing benefits. Fog-based VANETs consist of a series of mobile nodes that are fully dynamic without any central management. These networks have some limitations due to the use of the mobile power source. There are many challenges in extending mobile-based networks’ life span. Many methods have been suggested to minimize the nodes’ energy consumption and extend the network life. Also, some works have been developed for load-balanced routing. Still, energy-efficient routing and load balancing in VANETs are challenges. The purpose of this study is to present a suitable method based on energy awareness for load balancing in fog-based VANETs using a hybrid optimization algorithm (employing ant colony optimization and artificial bee colony (ACO-ABC)). The simulation results in the Network Simulator 2 (NS2) environment showed that as the number of nodes increased, the consumed amount of energy in the VANET is increased. Also, with the increasing number of tasks, load balancing by the proposed method has been improved. Finally, the simulation experiments showed that the proposed hybrid algorithm (ACO-ABC) outperforms all other algorithms.

1 | INTRODUCTION
A Vehicular Ad hoc NETwork (VANET) is a mobile network where its nodes are vehicles [1, 2]. The aim of VANETs is to allow vehicles to communicate with each other. Therefore, these nodes have to implement communication radio interfaces, and for VANET data transmission, a particular range spectrum should be allocated. Nodes require such attributes to assist them in collecting information, warning their neighbours, and making decisions by analysing all the information gathered to be an integral component of a VANET and interact effectively [3]. Recently, extending cloud computing to the network’s edge is a very hot topic, for example, deploying fog nodes to the network edge [4–6]. In other words, computational capabilities are pushed to devices located on the edge of the network based on the internet “decentralization” feature. It lets some services be decentralized from the cloud to the fog devices [7, 8], reducing data transmission delay, bandwidth consumption, computational costs etc. [9]. Furthermore, in the case of VANETs, fog computing can potentially provide the following [2]:

- Timely detection of dangerous driving behaviour, providing early warning and imposing appropriate punishment if needed;
- Real-time traffic control, for example, by changing the timing of the traffic lights when there is no approaching car;
- Real-time reminders can warn drivers to slow down in emergencies by controlling traffic lights using the monitored data.

A vehicle can conveniently acquire location information using global positioning system (GPS). An easy approach to remove network bottlenecks, maximize network throughput, and boost network flexibility is a load balancing approach. Nevertheless, it is hard to evaluate and optimize the network load of VANETs because of a lack of global network knowledge [10]. In addition, because of the semi-organized existence of...
vehicular movements subject to road layout restrictions and the barriers that restrict physical connectivity in urban environments, VANETs offer a specific set of challenges and chances for routing protocols [1].

Most research on various routing protocols in IoT and VANET recently focuses on mechanisms to create energy-efficient routes [11]. Energy plays a significant role in performing significant operations in VANET networks. As the node energy decreases, the network connection decreases as well. Node failure due to power limitation causes system failure; hence, it terminates the internet connection [12].

We are working to create a more effective hybrid VANET routing protocol for data transmission and load balancing objectives. Since this problem is an NP-Hard problem, the meta-heuristic and nature-inspired algorithms are the best choices for solving it. This paper presents a new energy-aware method for load balance managing in the fog-based VANET using a hybrid optimization algorithm. Several bio-inspired algorithms show their efficacy in load balancing systems, including an ant colony optimization (ACO) and artificial bee colony (ABC). Most of them lack decent performance in all aspects. Therefore, the benefits of each algorithm are used by hybrid algorithms [13]. In this paper, for discovering the initial solution collection, the ACO algorithm is utilized. We try to eliminate the ants' stagnation behaviour and the time-consuming global search for initial solutions by the employed bees. In the proposed algorithm, ants use the bees' exploitation to determine the best solution and best feature subset. Bees adapt the feature subsets generated by the ants as their food sources [14]. The ABC algorithm is also utilized to evaluate and strengthen each of the likely solutions given by the ACO module [15]. The main contributions of the paper are:

- Reducing total energy consumption of fog-based VANETs using ACO-ABC algorithm;
- Improving the load balance among nodes using ACO-ABC algorithm;
- Investigating the network lifespan by measuring the amount of residual energy;
- Investigating the life cycle of fog-based VANETs nodes using ACO-ABC algorithm.

The whole article consists of five sections. Section 1 contains the introduction and generalities. In Section 2, the previous conducted investigations and studies on this subject will be reviewed. Section 3 describes the various steps taken within the research. The research outcomes are presented in Section 4. Ultimately, general conclusions and future work are provided in Section 5.

2 RELATED WORK

Diverse algorithms for load balancing were led to a wide review of various literature. Bio-inspired algorithms are the most attention-grabbing ones. Several investigators studied the essence of bio-inspired to balance the load. Swarm intelligence algorithms and evolutionary algorithms are sub-categories of bio-inspired algorithms. They were utilized to solve many problems like load balancing and scheduling, aiming at optimization [13]. Evolutionary algorithms have been established to replicate the selection and enhancement of natural behaviour. Swarm intelligence-based algorithms rely on particular familiar living creatures’ actions, like bees, ants, fish, and birds, who have their unique means of getting the problem-finding space [16, 17]. Some significant relevant works in this area are listed in the remainder of this paper.

Murugan et al. [18] examined ACO and PSO’s performance in Mobile Adhoc NETWORKS (MANET) and VANET to transmit the data in the shortest route efficiently and evaluates energy consumption and load balancing among nodes. The performance analysis in this paper was done on the NS2 environment. They used some parameters such as successive packet delivery, transmission, packet drop, delay, Goodput, and dropping ratio. The results showed that using ACO and PSO selects the most reliable path, which reduces the possibility of link breakages, i.e. particular zone area, and responds better to changes in the network topology.

Elhoseny and Shankar [19] presented a K-Medoid clustering model to cluster the vehicle nodes. Afterward, for convincing communication, energy-efficient nodes are identified. A heuristic algorithm identifies effective nodes from each cluster with the anticipation of achieving energy-efficient contact to optimize the energy usage in VANET. The findings revealed that vehicle-to-vehicle (V2V) connectivity increases all vehicle nodes’ energy efficiency, thus achieving less implementation time relative to current algorithms.

Also, Sekar and Mangalam [20] presented self-generation and co-evolution based memetic optimization (SGC-MO) technique for energy-efficient routing stability and load balancing in MANET. The suggested SGC-MO approach preserves the energy-efficient path's stability and balances the load when routing the packets. It also assists in improving the distribution ratio of packets and the load balance factor throughout the procedure of load balancing. The simulation findings revealed that the SGC-MO methodology has improved efficiency with a load balancing factor gain of 22 percent and a packet distribution ratio of 15 percent relative to current approaches.

Panda, et al. [21] proposed a virtual machine migration technique (VMMT) based load balancing approach for the underloaded hosts. They compared their proposed method with the existing technique using the load imbalance level and energy usage. The results showed that the proposed approach provided a good load imbalance. Also, the energy consumption of the proposed model is lower than the existing model.

Abdukodir et al. [22] developed an edge computing distribution method in VANET-based real-time systems. They described VANET network architecture based on software defined network (SDN)/mobile edge computing (MEC) systems to reduce network load and traffic density. Also, they examined the possibility of temporarily placing the application to the restricted stock units (RSU) for reducing the load. The proposed architecture allows optimal use of the RSU/MEC resources and reduces the load and latency.
Furthermore, Panda, et al. [21] proposed a virtual machine migration technique (VMMT)-based load balancing approach for the under loaded hosts. They compared their proposed method to the existing technique using the load imbalance level and energy usage. The results showed that the proposed approach provided a good load imbalance. Also, the energy consumption of the proposed model is lower than the existing model.

Kout et al. [23] suggested a routing protocol inspired by the cuckoo search behaviour. For the mobility model, they used the random waypoint model. To validate their research, they compared the protocol to other protocols in terms of the quality-of-service parameters: packet delivery ratio and end-to-end delay (E2ED). The findings showed that their protocol is more powerful in terms of E2ED and packet delivery ratio (PDR) than other protocols.

Also, Maleki et al. [24] discussed the bi-objective issue of postponement and energy-efficient routing. To approximate the optimal routing policy of the formulated Markov decision problem (MDP), they also suggested a model-based reinforcement learning (RL) algorithm. The multi-agent basis of their suggested RL approach also facilitates the global optimization of the system. The findings revealed that their model-based method beats its model-free equivalent and fits similarly with traditional value-iteration, which takes perfect statistics into account.

Shukla, et al. [25] presented a composite VANET and cellular networks. With no additional network cost, they enhanced a routing technique for hybrid networks. Firstly, the high propagation speed of VANET and the large-scale cellular infrastructure are the advantages of this composite network. The network also dominates cellular and ad-hoc network disadvantages. The findings revealed that the routing algorithm reduced the propagation time and requested the block rate.

Babu et al. [26] suggested an ABC algorithm for efficient load balancing, which is relied on the foraging behaviour of honey bees to balance the load. Tasks omitted from overloaded VMs are viewed as honey bees in the proposed methodology, and the food supplies are under loaded VMs. The goals of tasks in VM waiting queues were also regarded, and efforts to reach minimal response time and a decreased number of task migrations were made. The findings demonstrated that the quality of service improved dramatically.

Finally, Jain and Singh [27] proposed an ACO-based approach for load balancing in a peer-to-peer grid environment. They used MATLAB software for simulation. The results showed that the ACO achieves a better optimal solution.

3 | METHODOLOGY

In this section, we first introduce the features of the VANET network and network routing protocols. Then, while introducing the ACO algorithm and ABC algorithm, the proposed method’s steps are stated. In addition, the routing protocol is stated, and at the end of this section, fitness is calculated.

3.1 | Characteristics of VANET networks

As follows, the VANET characteristics are described [28, 29]:

- VANET consists of a large number of vehicles, particularly within high traffic intensity periods.
- There are very precise and well-established priorities for the applications created for VANET, like the availability of secure and intelligent transport networks to provide valuable opportunities for continued work.
- Rather than demanding a particular facility or route [30], several VANET-related safety applications are seeking to provide traffic-related information to all available nodes in a specific geographical region.
- In the predefined road network, there are mobile nodes in the VANET. Consequently, this topology helps the direction of the vehicle to be ascertained. Additionally, because of barriers on separate but nearby highways, vehicles would potentially disconnect [31].
- Network disintegration cycles may occur throughout low traffic density periods.
- The complex topology of the network is because of the heavy flow of vehicles.

The network model of fog-based VANET is shown in Figure 1. Fog computing is superimposed on cloud computing in the VANET case. In order to solve data processing and storage services, it provides a distributed VANET component-computing network. The function of fog computing among the far-end servers and the front devices is the same as near-end computing proxies [32].

3.2 | Routing protocols in VANET networks

Routing protocols in VANET networks are divided into two categories: proactive and reactive.

3.2.1 | Proactive routing protocols

In this method, nodes exchange routing information (in order to keep this information accurate and consistent) alternately. In mobile networks, node motion creates a dynamic topology that may require repeated routing table constructions, while these tables themselves can be very large. The advantage of this type of protocol is that all nodes’ path is available at any time, and there is no need for a path discovery mechanism. The disadvantage of this type of protocol is its high overhead. The most popular basic protocol that uses this method is destination sequenced distance vector (DSDV). Optimized link state routing (OLSR) and wireless routing protocol (WRP) are the other popular proactive routing protocols [33].
3.2.2 | Reactive routing protocols

In reactive or demand-driven routing protocols, when a node does not know the node’s path, it initiates a path discovery mechanism for that node. These protocols are more efficient and contain less overhead than proactive protocols. It is because any change in the network topology will be understood immediately. Secondly, if no change is made, no overhead will be imposed on the system, and there will be no need to send intermittent route coordination packets. The implementation process of reactive routing protocols consists of two stages: path discovery and path maintenance. Ad-hoc on-demand distance vector (AODV) and dynamic source routing (DSR) are the most popular basic protocols based on this method [34].

3.3 | ACO and ABC algorithms

Given that the load balancing issue is an NP-hard problem, meta-heuristic methods can be used to solve it [35, 36]. This section’s primary purpose is to use a hybrid algorithm (ACO and ABC) to solve this problem. Swarm intelligence is a new and powerful approach to solve optimization issues relying on group behaviours [37]. An obvious example of this intelligence can be seen in the behaviour of insects and animals living as colonies. In this paper, a hybrid ACO-ABC algorithm is used. In the proposed method, first, an array of \(1 \times n\) of random numbers is generated to initialize the algorithm. This array indicates the number of nodes that can perform the tasks in the workflow. Next, a matrix is created that maintains the list of nodes that can perform the workflow tasks. Afterward, a prototype is generated, which is a \(1 \times n\) array of nodes. In this array, the node in the entry \(i\) performs the workflow’s \(i\)th task. After creating the sample and calculating its objective function, this sample is selected as the solution, and the algorithm moves towards it. The next step is to create a list of suggested nodes to perform the tasks in the workflow. Each provided workflow by the user has \(m\) tasks. There are a determined number of nodes to perform each of these tasks. For instance, in Figure 2, the workflow has three tasks. Figure 2 shows the number of nodes suggested for each of the tasks in the workflow. As shown in Figure 3, for the first task, \(p\) node is proposed, for the second task, \(q\) node is proposed, and for the third task, \(z\) node is proposed. In the next step, a list of suggested nodes for each task will be created. In the proposed method, the number of nodes proposed for each task is randomly generated.

The following is a list of nodes that perform each task. In this study, the number of nodes is assumed to be 10, 20, 30, 40, and 50. The nodes were randomly scattered in different parts of the environment. Hypothetical paths are then considered for the respective nodes. Table 1 shows the list of nodes that can perform each of the tasks.

- **Artificial bee algorithm**

The ABC is an artificial intelligence method inspired by the feeding behaviour of bees [38]. A colony of bees in nature
consists of three parts: food sources, employer bees, and non-employer bees. Worker bees only communicate with the food source from which they are currently extracting nectar. Besides, these bees carry information such as distance, direction, and fitness of the resource and share this information with others in the hive. Non-employer bees, on the other hand, are always looking for food sources to extract their nectar [39]. Non-employer bees are divided into two main groups: scout bees and onlooker bees. Scout bees explore the can find new food sources by searching the environment, and onlooker bees wait in the hive to receive information from employer bees. Honey bees can obtain information about the location and quality of food sources outside the hive. Employer bees share their knowledge by dancing with the bees in the hive (onlooker bees). Communication between bees is done through the language of dance. The employer bee exhibits a series of movements in different parts of the hive, dancing area, to inform the bees in the hive and to share the information obtained about the value, location, and distance of the food source from the hive. The food source’s position and distance, the angle of the food source to the sun, and the food source’s value are determined according to the different types of dances. In general, any employer bee can do the following three dances to share its information [40].

- Round dance
  It does not provide information about the direction of the food source; employer bees perform this dance when the food source is close to the hive.
- Waggle dance
  With this dance, employer bees provide information to other bees about the direction of the food source according to the position of the sun. Employer bees perform this dance when the food source is away from the hive, and the dance speed is proportional to the distance the hive is from the food source.
- Tremble dance
  If the employer bee spends a lot of time extracting and exploiting the food source, it does this dance. This dance shows that the bee has no information about its food source’s current profitability and has spent a lot of time extracting nectar from that source. Employer bees perform this dance when the limit parameter is violated.

The ABC algorithm is utilized in numerous fields to solve diverse optimization issues [41, 42]. The control parameters, operators, and utilization fields of the ABC algorithm are seen in Figure 4. Some fields include routing, timing issues, image segmentation, redundancy allocation etc.

The employed bee is searching around the source in the utilized ABC algorithm. Each of them goes from one old position \( x_j \) to novel candidate one \( v_j \) utilizing Equation (1) [44].

\[
\begin{align*}
\nu_j &= x_j + \phi_j (x_j - x_k) \\
\end{align*}
\]

where \( k \in \{1, 2, \ldots, SN\} \) and \( j \in \{1, 2, \ldots, D\} \) have been accidentally selected where \( SN \) is the number of food sources and \( D \) denotes the dimension of the issue. According to Equation (1), \( k \) should be various from \( i \), and \( \Phi_j \) is a uniform random number in the range of \([−1, 1]\). If the novel position value \( v_j \) is higher than \( x_j \), it is updated. Utilizing a roulette wheel selection process like genetic algorithm [45], the onlooker bee chooses a food source using the probability value, and then this novel location is computed using Equation (2).

\[
\begin{align*}
\nu_j &= x_j + w_i \phi_j (x_j - x_k) \\
\end{align*}
\]

where \( w_i \) is the weight coefficient of employed bee information. The probability of the food source can be obtained using [46]:

\[
\begin{align*}
\rho_i &= \frac{fit_j}{\sum_{j=1}^{SN} fit_j} \\
\end{align*}
\]

where \( fit \) denoted the fitness values of the food sources of employed bees, and it is defined by Equation (4).

\[
\begin{align*}
fit_i &= \begin{cases} 
\frac{1}{1 + f(x_i)} & f(x_i) \geq 0 \\
1 + \frac{1}{f(x_i)} & f(x_i) < 0 
\end{cases}
\end{align*}
\]

\( f(x_i) \) shows the amount of the objective function values.

- Ant colony algorithm

The ants-based algorithm, which has been inspired by many techniques and has had the greatest success in optimization techniques, is known as the ACO algorithm. The ACO algorithm is one of the meta-heuristic methods [47]. Ant’s food search behaviour inspires ACO. There is no direct communication between the members, and they only communicate with each other indirectly through signs. Ants leave traces of the pheromone chemical as they walk. The substance evaporates soon but remains on earth for a short time as an ant trail. There is a simple basic behaviour in ants. When choosing between two paths, they statistically choose a path that has more pheromones, or in other words, more ants have already passed through [48]. This is a simple preparation to find the shortest path and this process can be described as a positive feedback loop. It is important to note that although ants statistically choose the pheromone pathway, this is still possible and uncertain. The evaporation of the pheromone and the possibility of an accident allow the ants to find the most optimal path. These two features make it flexible in solving any optimization problem [49]. This algorithm was first introduced to solve

**TABLE 1**  Suggested nodes for each of the tasks in the workflow

| List of the nodes for task 1 | List of the nodes for task 2 | List of the nodes for task 3 |
|-----------------------------|-----------------------------|-----------------------------|
| \( p_1 \)                  | \( q_1 \)                  | \( z_1 \)                  |
| \( p_2 \)                  | \( q_2 \)                  | \( z_2 \)                  |
|                           |                            |                            |
| \( p_n \)                  | \( q_n \)                  | \( z_n \)                  |
discrete problems such as traveling salesman, routing, and scheduling. In these issues, the variable is characterized by a finite set of components. ACO has achieved great success in the field of discrete optimization. Although the colony was first proposed for discrete optimization problems, its use to solve continuous optimization problems was then considered. In this set of optimization problems, the variable takes a real value from the defined domain. To implement ant colony, artificial ants are used as elements in optimization. The differences between them and real ants are:

- Memory: A memory can be considered for artificial ants that holds pathways;
- Artificial obstacles: Changing the details of the problem to test the algorithm and attain various answers;
- Life in a discrete environment: Real ants cannot survive apart from a colony.

The control parameters, operators, and application areas of the ACO algorithm are represented in Figure 5. The operators of the ACO algorithm are pheromone trail update and evaporation. The control parameters utilized by the ACO algorithm contain the ants’ and iterations’ number supposed in the algorithm. In Figure 5, some application fields and features where the ACO algorithm can be utilized are shown.

However, the pheromone values are updated at the end of each iteration [50].

\[
P_{ij}^k = \begin{cases} 
\frac{(\eta_j)^{\alpha} (\eta_i)^{\beta}}{\sum_{m \in N_j^k} (\tau_{im})^{\alpha} (\eta_m)^{\beta}} & j \in N_i^k \\
0 & \text{otherwise}
\end{cases}
\]

where \( P_{ij}^k \) is the probability to move from node \( i \) to node \( j \) by ant \( k \). This decision depends on the pheromone level and heuristic information. While \( N_j^k \) is the set of feasible neighbourhoods that have not yet been visited by ant \( k \), \( \eta_{ij} \) is a heuristic function, and \( \tau_{ij} \) is the amount of pheromone on edge \( i \) and \( j \), and \( \alpha \) and \( \beta \) are the parameters that control the relative importance of pheromone concentration and heuristic information. The pheromone update can be formed as follows:

\[
\tau_{ij} \leftarrow \tau_{ij} + \Delta \tau_{ij}^k
\]

\[
\Delta \tau_{ij}^k = \begin{cases} 
\frac{Q}{f(\psi^k)} & j \not\in \psi^k \\
0 & \text{otherwise}
\end{cases}
\]

The evaporation update is given by:

\[
\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}
\]

where \( \rho \) is the constant factor reduction of all pheromones, \( f(\psi^k) \) is the cost of the solution which is performed by ant \( k \), and \( Q \) is a constant. The above optimization process is ended after a certain amount of iteration.

- Hybrid ACO-ABC

Among the priory stated optimization procedures, ACO has absorbed great attention in the discrete issue domains because of its population-based search capability, simplicity, and robustness. To generate a good initial solution, ACO used the heuristic method and determined a good search direction based on the
experience. It is noteworthy that this approach also allows ACO to make a suitable solution and causes ACO to be trapped in local minima.

The ACO allows the rapid discovery of suitable solutions. Distributed computations prevent premature convergence. In this algorithm, convergence is guaranteed, but convergence time is not guaranteed (slow convergence speed). The ACO algorithm outperforms other algorithms for both global and local search modes. This algorithm has a higher execution speed and accuracy. Therefore, the probability of finding the optimal solution increases significantly. The artificial bee colony optimization algorithm, like most algorithms, has disadvantages such as stuck in the local optima. In some cases, the convergence speed of this algorithm is debatable. As the population grows, the computational cost increases. The artificial bee algorithm converges rapidly.

The hybrid ACO-ABC algorithm was expected to solve these drawbacks and incorporate both ACO and ABC’s merits. The optimization of the ant colony hybridization with the ABC algorithm takes advantage of ACO and ABC by dividing the optimization issue into two parts. Ant constructs the candidate solution in each iteration to place the load balance; after that, ABC optimizes the loads’ size. There are several typical features of the suggested algorithm. First, in order to escape from the local minimum solution, the triple search capability of the ABC algorithm is utilized. It causes ACO to affect the search process since it can rapidly find the right global optimum. Second, by dividing the optimization problem into continuous and discrete category stages, the algorithm’s performance is enhanced; thus, the search space’s size is decreased. Eventually, the classical ACO algorithm’s disadvantage, which is not appropriate for continuous optimization, can be resolved.

The flowchart of the proposed ACO algorithm and ABC optimization algorithm in energy-aware load balancing management is shown in Figure 6.

3.4 | Routing protocols based on load balance

So far, several load balancing routing protocols have been proposed. Most of these methods are based on integrating load balancing strategy with path discovery. The problem of load congestion often occurs at low bandwidth. Failure to do so will increase latency, reduce missing packet rates, end-to-end latency, and reduce battery power consumption. Therefore, it is necessary to balance the load in such cases. The main purpose of load balancing protocols is to divert data traffic from paths and nodes around congestion or nodes that receive larger data than other nodes.

Load balancing routing is generally performed based on latency, traffic, or in a hybrid manner. In the first one, load balancing is used to prevent the node with high link delay. In the traffic method, the equilibrium load balancing is evenly distributed among the obtained traffic nodes. In the last one, load balancing is obtained as a combination of traffic characteristics and latency. Load balancing nodes exist in load balancing routing protocols. These nodes have the high processing power, and the relevant paths can lead to these nodes. These nodes work by identifying low-congestion nodes and diverting routing from high-congestion nodes to low-congestion ones.
3.5 | Proposed routing protocol

In this research, the routing protocol is proposed by considering the node’s residual energy. This protocol is derived from the article [23]. When the average load on existing links increases beyond a threshold and the node’s remaining power falls below the threshold, data transfer to the corresponding node is prevented.

The energy used to transmit the request to send (RTS) messages from the source node to the destination node is obtained from Equation (11).

\[
RTS = \lambda \times d_{\text{max}}^R \times t_{\text{RTS}} E_{\text{RTS}}^{\lambda} \tag{11}
\]

- \( \lambda \): Data frequency
- \( d_{\text{max}}^R \): Maximum transmission range
- \( R \): Missed power of the path
- \( t_{\text{RTS}} \): RTS message transmission time

The energy consumed for CTS messages from destination to source is derived from Equation (12).

\[
CTS = \lambda \times d_{\text{max}}^R \times t_{\text{CTS}} E_{\text{CTS}}^{\lambda} \tag{12}
\]

The following two conditions are met for better path discovery.

- Path with minimum total energy consumption \( E_{\text{tot}} \)
- Path with high residual battery power \( b^f \)

Initially, paths to discovery are assumed.

The total energy \( (E_{\text{tot}})\) used in the network to transmit RTS and CTS messages is calculated by Equation (13).

\[
E_{\text{tot}} = d_{i}^{\text{RTS}} + d_{i}^{\text{CTS}}, \quad i = 1, 2, 3, 4, ... \tag{13}
\]

The residual energy in each path is calculated based on the power consumed (Equation (14)).

\[
D_{i}^{\text{RTS}} = \sum_{j=1}^{N} b_j, \quad i = 1, 2, 3, 4, ... \tag{14}
\]

3.6 | Fitness calculation

Here, to calculate the fitness, the total energy value \( (E_{\text{tot}})\) and the load deviation are calculated and considered fitness. The fitness
TABLE 2  Simulation parameters

| Parameter                  | Amount                                      |
|----------------------------|---------------------------------------------|
| Simulation environment size | 2000 × 2000 m²                               |
| Simulation time            | 100 s                                       |
| Medium capacity            | 6 Mbps                                      |
| PHY/MAC layer              | 802.11 p IEEE                               |
| Message transmission range | 1000 m                                      |
| Transfer layer             | UDP                                         |
| CBR packet size            | 100 bit                                     |
| CBR time                   | 60 s                                        |
| The number of nodes        | 10, 20, 30, 40, 50                           |
| Initial energy             | 0.5–1 J                                     |

function was calculated as follows:

\[ F = W_1 (E_{tot}) + W_2 \text{ (Deviation of the load)} \]  \hspace{1cm} (15)

where \[ W_1 + W_2 = 1 \]. The objective function mentioned above should be normalized as follows:

\[ \text{Normative} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \]  \hspace{1cm} (16)

Thus, total energy consumption and load deviation were calculated for all assumed nodes, and nodes with lower total energy consumption and better load balancing were considered.

4  RESULTS

In this section, the relevant results are presented.

4.1  Introducing the simulation environment and simulation parameters

NS2 simulator software is used for simulation in this article. The NS2 simulator is one of the most popular open-source network simulators. It is used for network research, which is a discrete event simulator. The NS2 simulator is the second version of the NS simulator. NS is basically based on a network simulator called REAL. Also, the simulation parameters are presented in Table 2.

4.2  Simulation results

For investigating the convergence of the proposed algorithm, the objective function is plotted in the considered iterations. The results show that the proposed method has better fitness than the ACO algorithm [27], the ABC algorithm [26], the VMMT [21] and Abdukodir et al. [22]. In addition, the algorithm’s stability test with 45 executions shows the excellent stability of the proposed algorithm. To test the convergence, the proposed algorithm is compared to the ACO, ABC algorithms, VMMT and Abdukodir et al. [22] in 110 replications. Figure 7 shows the convergence test for 100 tasks with 50 candidate nodes for each task in 110 repetitions.

4.2.1  Total energy consumption by the network

Figure 8 shows a diagram of the total energy consumed by the network in the three examined algorithms. The number of nodes is 10, 20, 30, 40, and 50. The diagram shows the good performance of the proposed method.

4.2.2  Network lifespan

To check the lifespan of the network, the amount of residual energy was measured. The more energy residual, the longer the network will last. The results mentioned in Figure 9 show that as the number of nodes increases, the amount of residual energy decreases, so the battery lifespan decreases.

4.2.3  Network life cycle

To examine the network’s life cycle within the period, the number of sensor nodes within the period was examined, the number of nodes being 50 candidate nodes with 100 tasks. The results mentioned in Figure 10 show that as the number of cycle’s increases, the number of residual nodes decreases.
Imbalance degree

The degree of imbalance measures among nodes is computed using Equation (15) [51].

\[ DI = \frac{T_{\text{max}} - T_{\text{min}}}{T_{\text{avg}}} \]  

(15)

where \( T_{\text{max}} \) is the maximum, and \( T_{\text{min}} \) is the minimum amongst all nodes, and \( T_{\text{avg}} \) is calculated by Equation (16).

\[ T_{\text{avg}} = \frac{\text{total tasks length}}{\text{number of nodes}} \]  

(16)

Figure 11 shows each algorithm’s average degree of imbalance when the number of tasks varying from 100 to 500. It can
be seen that the ACO-ABC can do superior system load balancing than ACO, ABC algorithms, VMMT and Abdukodir et al. and guarantees the load balancing in a large number of tasks.

5 CONCLUSION

The vehicular fog has recently been suggested to process big data and complex, intelligent analysis of VANET settings. Fog computing can be implemented to alleviate the computing and communication load on the vehicular cloud. By adding an intermediary layer between mobile devices and the cloud, fog computing is an expansion of the cloud-based Internet, aimed at the seamless, low-latency transmission of service from cloud to mobile. In this study, utilizing a hybrid optimization algorithm, we introduced a novel energy-aware technique for load balance management in the fog-based VANET.

A VANET network is made up of mobile nodes without any infrastructure. The mobility of nodes, limited energy reserves, lack of central management, and providing a guarantee for the quality of services are some of the challenges of this type of network compared to wired networks. The existing nodes of VANET networks are free to move in any direction. This study aimed to present an energy-aware method based on load balancing using the ACO algorithm. Algorithm implementation and routing methods were fully described. Then, the effect of increasing the number of nodes on the amount of energy consumed by the fog-based VANET and the residual energy in the battery were investigated. Also, the number of residual nodes was increased by increasing the length of routing periods in the proposed method with the ant and ABC optimization algorithm.

The simulation results in the NS2 environment showed that with increasing the number of nodes, the amount of energy consumed by the fog-based VANET increases. Also, as the number of nodes increases, the amount of residual energy will decrease, making the proposed algorithm outperform the other two algorithms. Examining the amount of residual energy within the periods revealed that the amount of residual energy decreased with increasing periods. The results indicate the good performance of the proposed method compared to the other four methods.

ACKNOWLEDGMENTS

This work was supported by the Anhui Province University Excellent Young Talents Support Project (No. GXYQ2019119), the Key Natural Science Projects of Anhui Province (No. KJ2018A0820) and Bozhou University Scientific Research No. (BYZ2018C05).

REFERENCES

1. Cooper, C., et al.: A comparative survey of VANET clustering techniques. IEEE Commun. Surv. Tutorials 19(1), 657–681 (2016)
2. Ma, M., et al.: An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks. IEEE IoT J. 6(5), 8065–8075 (2019)
3. Engoulou, R.G., et al.: VANET security surveys. Comput. Commun. 44, 1–13 (2014)
4. Bonomi, F., et al.: Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, pp. 13–16 (2012)
5. Kashani, M.H., Rahmani, A.M., Navimipour, N.J.: Quality of service-aware approaches in fog computing. Int. J. Commun. Syst. 33(8), e4340 (2020)
6. Darbandi, M.: Proposing new intelligent system for suggesting better service providers in cloud computing based on Kalman filtering. HCTL Int. J. Technol. Innovations Res. 24(1), 1–9 (2017)
7. Huang, H., et al.: Bitcoin-based fair payments for outsourcing computations of fog devices. Future Gener. Comput. Syst. 78, 850–858 (2018)
8. Darbandi, M.: Kalman filtering for estimation and prediction servers with lower traffic loads for transferring high-level processes in cloud computing. Int. J. Technol. Innovations Res. 23(1), 10–20 (2017)
9. Wang, L., Liu, G., Sun, L.: A secure and privacy-preserving navigation scheme using spatial crowdsourcing in fog-based vanets. Sensors 17(4), 668 (2017).
10. Wu, D., et al.: Geographic load balancing routing in hybrid vehicular ad hoc networks. In: 2011 14th International IEEE conference on intelligent transportation systems (ITSC). Washington, DC, pp. 2057–2062 (2011)
11. Sadrinojajaei, M., et al.: A new preventive routing method based on clustering and location prediction in the mobile Internet of Things. IEEE IoT J. 1–1 (2021)
12. Sharma, R.K., et al.: Multicriteria based intelligent energy efficient routing scheme using spatial crowdsourcing in fog-based vanets. Sensors 17(4), 668 (2017).
13. Gamal, M., et al.: Osmotic bio-inspired load balancing algorithm in cloud computing. IEEE Access 7, 42735–42744 (2019)
14. Shunmugapriya, P., Kammani, S.: A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). Swarm Evol. Comput. 36, 27–36 (2017)
15. Sen, T., Mathur, H.D.: A new approach to solve economic dispatch problem using a hybrid ACO–ABC–HS optimization algorithm. Int. J. Electr. Power Energy Syst. 78, 735–744 (2016)
16. Balusamy, B., et al.: Bio-inspired algorithms for cloud computing: A review. Int. J. Innovative Comput. Appl. 6(3–4), 181–202 (2015)
17. Aghajani, G., Ghadimi, N.: Multi-objective energy management in a microgrid. Energy Rep. 4, 218–225 (2018)
18. Murugan, S., et al.: Comparison of ACO and PSO algorithm using energy consumption and load balancing in emerging MANET and VANET infrastructure. J. Crit. Rev. 7(9), 2020 (2020)
19. Elhoseny, M., Shankar, K.: Energy efficient optimal routing for communication in VANETs via clustering model. In: Elhoseny M., Hassanien A. (eds) Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks. Studies in Systems, Decision and Control, vol 242. pp. 1–1 (2021)
20. Sekar, P.C., Mangalam, H.: Third generation memetic optimization technique for energy efficient routing stability and load balancing in MANET. Cluster Comput. 22(5), 11941–11948 (2019)
21. Panda, B., et al.: Energy aware virtual machine consolidation for load balancing in virtualized environment. In: 2019 International Conference on Communication and Electronics Systems (ICCES). Coimbatore, India, pp. 180–185 (2019)
22. Abdukodir, K., et al.: Development of edge computing distribution method in VANET based real-time systems. In: 2019 IEEE International Conference on Electrical Engineering and Photonics (EE&Politech). St. Petersburg, Russia, pp. 120–123, IEEE (2019)
23. Kout, A., Labeed, S., Chikhi, S.: AODVCS, a new bio-inspired routing protocol based on cuckoo search algorithm for mobile ad hoc networks. Wireless Networks 24(7), 2509–2519 (2018)
24. Maleki, M., Hakami, V., Dehghan, M.: A model-based reinforcement learning algorithm for routing in energy harvesting mobile ad-hoc networks. Wireless Pers. Commun. 95(3), 3119–3139 (2017)
25. Shukla, R.S., Maurya, D., Maurya, B.: Data dissemination under load distribution in hybrid network for VANET. In: 2016 International Conference System Modeling & Advancement in Research Trends (SMART). Moradabad, India, pp. 175–181 (2016)
26. Babu, K.R., Joy, A.A., Samuel, P.: Load balancing of tasks in cloud computing environment based on bee colony algorithm. In: 2015 Fifth
27. Jain, A., Singh, R.: An innovative approach of Ant Colony optimization for load balancing in peer to peer grid environment. In: 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). Ghaziabad, India, pp. 1–5 (2014)

28. Domingo, M.C., Reyes, A.: A clean slate architecture design for VANETs. Wireless Pers. Commun. 67, 315–333 (2012)

29. Jarupan, B., Ekici, E.: A survey of cross-layer design for VANETs. Ad Hoc Networks 9(5), 966–983 (2011)

30. Li, M., Zeng, K., Lou, W.: Opportunistic broadcast of event-driven warning messages in vehicular ad hoc networks with lossy links. Comput. Networks 55(10), 2443–2464 (2011)

31. Arshad, M., et al.: A survey of local/cooperative-based malicious information detection techniques in VANETs. EURASIP J. Wireless Commun. Networking 2018(1), 62 (2018)

32. Grover, J., et al.: Real-time vanet applications using fog computing. In: Somani A., Srivastava S., Mundra A., Rawat S. (eds) Proceedings of First International Conference on Smart System, Innovations and Computing. Smart Innovation, Systems and Technologies. vol. 79. pp. 683–691. Springer, Singapore (2018)

33. Dhenakaran, S., Parvathavarthini, A.: An overview of routing protocols in mobile ad-hoc network. Int. J. Adv. Res. Comput. Sci. Software Eng. 3(2), (2013)

34. Gorantala, K.: Routing protocols in mobile ad-hoc networks. Master’s Thesis. Umeå University, Sweden (2006)

35. Akbar Neghabi, A., et al.: Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network. Int. J. Commun. Syst. 32(4), e3875 (2019)

36. Neghabi, A.A., et al.: Energy-aware dynamic-link load balancing method for a software-defined network using a multi-objective artificial bee colony algorithm and genetic operators. IET Commun. 14(18), 3284–3293 (2020)

37. Naseri, A., Navimipour, N.J., A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J. Ambient Intell. Hum. Comput. 10(5), 1851–1864 (2019)

38. Li, D.B., Krishna, P.V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Comput. 13(5), 2292–2303 (2013)

39. Sharifi, S., et al.: Environmental economic dispatch using improved artificial bee colony algorithm. Evol. Syst. 8(3), 233–242 (2017)

40. Kiran, M.S., Gunduz, M.: A novel artificial bee colony-based algorithm for solving the numerical optimization problems. Int. J. Innovative Comput. Inf. Control 8(9), 6107–6121 (2012)

41. Firouz, M.H., Ghadimi, N.: Concordant controllers based on FACTS and FPSS for solving wide-area in multi-machine power system. J. Intell. Fuzzy Syst. 30(2), 845–859 (2016)

42. Gollou, A.R., Ghadimi, N.: A new feature selection and hybrid forecast engine for day-ahead price forecasting of electricity markets. J. Intell. Fuzzy Syst. 32(6), 4031–4045 (2017)

43. Rajan, C., et al.: Investigation on bio-inspired population based meta-heuristic algorithms for optimization problems in ad hoc networks. Int. J. Math. Comput. Phys. Electr. Comput. Eng. 9(3), 163–170 (2015)

44. Kefayat, M., Ara, A.L., Niazi, S.N.: A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources. Energy Convers. Manage. 92, 149–161 (2015)

45. Hamian, M., et al.: A framework to expedite joint energy-reserve payment cost minimization using a custom-designed method based on mixed integer genetic algorithm. Eng. Appl. Artif. Intell. 72, 203–212 (2018)

46. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. J. Global Optim. 39(2), 459–471 (2007)

47. Asghari, S., Navimipour, N.J.: Resource discovery in the peer to peer networks using an inverted ant colony optimization algorithm. Peer-to-Peer Networking Appl. 12(1), 129–142 (2019)

48. Asghari, S., Navimipour, N.J.: Cloud service composition using an inverted ant colony optimisation algorithm. Int. J. Bio-Inspired Comput. 13(4), 257–268 (2019)

49. Azad, P., Navimipour, N.J.: An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm. Int. J. Cloud Appl. Comput. 7(4), 20–40 (2017)

50. Niknam, T.: A new approach based on ant colony optimization for daily Volt/Var control in distribution networks considering distributed generators. Energy Convers. Manage. 49(12), 3417–3424 (2008)

51. Chien, N.K., Son, N.H., Loc, H.D.: Load balancing algorithm based on estimating finish time of services in cloud computing. In: 2016 18th International Conference on Advanced Communication Technology (ICACT). PyeongChang, Korea (South), pp. 228–233 (2016)

How to cite this article: Qun, R., Arefzadeh, S.M.: A new energy-aware method for load balance managing in the fog-based vehicular ad hoc networks (VANET) using a hybrid optimization algorithm. IET Commun. 15, 1665–1676 (2021). https://doi.org/10.1049/cmu2.12179