Optimizing Resource Allocation for Vehicular Edge Computing: A Hybrid Approach for Enhanced Efficiency and Task Management

[1]V. S. Nici Marx, [2]Dr. G. Ramachandran, [3]A. Kanthimathinathan

[1]Research Scholar, Dept. of Computer Science and Engineering
Annamalai University
Annamalainagar – 608002
Tamilnadu, India
[2]Associate Professor, Dept. of Computer Science and Engineering
Annamalai University
Annamalainagar – 608002
Tamilnadu, India
[3]Associate Professor, Dept. of Computer Science and Engineering
Annamalai University
Annamalainagar – 608002

Email: [1]nicimarx4896@gmail.com, [2]gmrama1975@gmail.com, [3]kanthi_88@yahoo.co.in

Abstract: In this research, present a hybrid optimized resource allocation model that takes into account multiple factors, including the status of available resources, geographical distances, and available network bandwidth, and the specific requirements of tasks to be executed with in Vehicular Edge Computing (VEC). The operational flow of the model begins with network formation, creating the foundation for effective communication and resource sharing. Feature extraction techniques are then applied to obtain relevant information from the available resources in the VEC domain. Following feature extraction, a detailed analysis identifies all connectable vehicles within the VEC. Feature selection processes are underpinned by the combination of the Walrus Optimization Algorithm and the Osprey Optimization Algorithm (WaOA-OOA) which refine the design, enabling a more efficient allocation. The core of the model involves the computation of a task-vehicle cost-time matrix, incorporating factors such as resource capabilities and task requirements. This matrix guides the subsequent vehicle clustering process, which utilizes the Farthest First K-means algorithm to form clusters of vehicles with similar resource profiles. The task allocation phase optimally assigns tasks to vehicle clusters, ensuring efficient resource utilization and minimal task delay. The Drawer Algorithm (DA) is proposed to optimize resource allocation by minimizing task completion times. In this algorithm, vehicles are grouped for each task, and the goal is to determine the most efficient way to assign subtasks to vehicles while minimizing the overall task completion time. The algorithm uses a cost-delay matrix to represent the time it takes for each subtask to be completed on each candidate vehicle where the key idea is to divide the subtasks among the vehicles in a way that optimizes the system's overall performance, which is formulated as a 0-1 integer linear programming problem. The proposed model seamlessly integrates with cloud servers to securely save allocation results, making them accessible for subsequent task execution. Simulation results underscore the effectiveness of our proposed method, revealing significant reductions in vehicle power consumption while consistently meeting task delay requirements. This research advances the resource utilization in both cloud and edge computing domains, and advances the state of resource allocation in VEC.
Keywords: Walrus Optimization Algorithm and the Osprey Optimization Algorithm (WaOA-OOA), Vehicular Edge Computing (VEC), feature selection, clustering, resource allocation, Farthest First K-Means algorithm

1. Introduction

The theory of cloud computing has made it possible to take use of surplus processing capacity. We will approach the large number of automobiles on streets, roads, and parking lots as abundant and underused computing resources that may be applied to the provision of public services [1,2]. Numerous cars spend hours each day in a driveway, parking lot, or garage. The automobiles that are parked represent a substantial untapped asset that is now being squandered. Vehicles are ideal candidates to be nodes in a cloud computing network because of these properties [3]. Three levels are the foundation of the Vehicular Cloud Computing (VCC) architecture: inside-vehicle, communication, and cloud. A collection of mobile cars interacting with one another over a wireless network is known as a VANET (Vehicular Ad-Hoc Network). Over the last several years, a great deal of study has been focused on this topic in an attempt to enhance traffic management and driving safety [4, 5]. New smart automobiles allow Internet connectivity and provide various services to drivers and passengers based on this advancement. Consequently, driving is safer, more pleasurable, and more comfortable. Onboard capabilities are still underused, albeit [6]. to make use of these resources, which include different location-specific services and online games.

Compared to VANET, VCC is a paradigm change, especially in light of mobile devices' limited CPU capacity. These restrictions often make it more difficult to submit data in real time. In contrast, VCC uses Cloud Computing (CC) [7–10] to provide a new method for drivers. As an alternative to forcing them to make investments in infrastructure and resource acquisition, VCC provides a pay-as-you-go strategy. This relieves drivers of the ownership responsibility and allows them to utilize computer resources as required. Numerous benefits come with a pay-as-you-go strategy, including quick access, high scalability, reduced operational costs, and lower investment [11]. This strategy allows users to utilize the required resources as much as their applications need without requiring an upfront investment or the installation and upkeep of software. One example of this kind of platform is IBM Smart-Cloud [12]. High-potential vehicles may share their apps and services at a reduced cost because to VCC technology. Put differently, cars that have substantial computing and storage resources may share them with other vehicles thanks to VCC [13, 14].

Several studies have been conducted to solve the important problem of resource allocation in VCC that is energy-conscious. In order to achieve optimal and efficient resource allocation, these investigations have used a wide range of algorithms, such as the Cuckoo Search Algorithm (CSA) [15] and the Whale optimization algorithm (WOA), Artificial Neural network (ANN), Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) algorithm, along with a multitude of techniques [16–18]. By distributing processing, storage, and networking resources across networked cars in an efficient manner, these algorithmic techniques aim to lower energy usage and improve overall VCC efficiency. These advanced algorithms optimize resource use and adjust to constantly shifting traffic patterns by using past data and real-time input to guide their judgments. Additionally, a concentrated effort has been made to look into safe and private resource allocation procedures in response to the urgent privacy and security issues, guaranteeing the safety of sensitive data shared inside the VCC environment. As long as academics continue to investigate and improve these algorithms, the promise of VCC to improve vehicle-to-vehicle communication and services will eventually come to pass. At the nexus of cloud computing and vehicular networking, this topic represents a vibrant and developing area of study [19, 20].

Contribution of the Research

- ❖ The research introduces a novel approach to resource allocation in multi-cloud environments by employing the Walrus Optimisation Algorithm (WaOA) and the Osprey Optimisation Algorithm (OOA)
- ❖ The Farthest First K-Means algorithm effectively chooses CHs and assembles clusters utilising node degree, distance, bandwidth and residual energy as input parameters.
- ❖ Improving network efficiency with regard to package transfer speed and delay reduction.
- ❖ Simulation results demonstrate the potential of the suggested method to reduce energy consumption in vehicular environments while meeting the requirements for minimizing vehicle delay.
- ❖ The performance of the proposed technique is analysed with various metrices using MATLAB platform. The proposed method is compared with the conventional techniques.

The remains of the article are structured as follows. Related work is included in Section 2. The proposed work's technique is presented in Section 3. The performance of the proposed approach is outlined in Section 4. The intended work is concluded in Section 5.

2. Literature Review

Building the cloud of RSUs and/or the hybrid cloud are the standard methods for obtaining the generality of the whole network, which guarantees the effective communication in VANETs. Many previous studies on the different clouds in VANETs will be briefly covered in this section.

NAUTILUS, a bat-bioinspired system for resource allocation in vehicular clouds, has been described by Quessada et al. [21]. In order to define pseudo-optimal decision-making for the allocation process in a vehicular cloud, the method optimizes the search process using the metaheuristic. In order to support the suggested method in the allocation process, they additionally take into account a fog-based paradigm that distribute the runtime, memory, processing, storage, and runtime from the vehicles. Two alternative algorithms that make use of conventional search methods were contrasted with the NAUTILUS algorithm: the Greedy approach and the Analytic Hierarchy Process (AHP) approach.

A Mobile Edge Computing (MEC) enabled Unmanned Aerial Vehicle (UAV) aided VANET architecture has been studied by He et al. [22]. This architecture enables UAVs with compute resources to service multiple vehicles. Due to limited computer power, each vehicle must delegate its computing chores to the appropriate MEC server aboard the UAV. In order to address the aforementioned issues, we first develop and examine the job computation model of the local vehicle and the edge UAV, as well as the transmission model and security assurance model from the vehicle to the MEC server on UAV. After that, the task offloading, resource allocation, and security assurance are all taken into consideration together while formulating the vehicle offloading issue as a multi-objective optimization problem.

A dynamic resource management approach for optimizing service capability in fog-enabled vehicle ad hoc networks has been studied by Thanedar et al. [23]. The issue is reduced from Seminar Assignment issue to establish its NP-hardness. The problem is represented as Integer Linear Programming (ILP). By representing the issue as a graph with vertices representing the Fog Nodes (FNs) and edges representing the cars present in the overlapping area of the pairs of FNs, a polynomial-time approach is shown. The suggested algorithm interacts with the associated FNs by taking into account the set of cars that are in overlapping coverage zones of FNs. Next, in order to reduce the allotted RBs, it migrates the Resource Blocks (RBs) of the set of cars between pairs of FNs.

A multi-objective method based on deep learning algorithms and differential evolution has been described by Taha et al. [24] for VANETs. Here, the cluster algorithm uses the Kubernetes container-base to choose different cars that meet the requirements of the algorithm. As a result, we are able to carry out difficult operations for data owner automobiles. Our method uses a deep learning model to determine the fit complexity of sub-tasks, and the cars' information is accessible on the master vehicle (data owner vehicle) when the vehicle enters the cluster. The MOTD-DE that has been suggested divides up subtasks among groups of cars in order to minimize task execution time and the number of vehicles required to complete it. Additionally, we take the subtasks to be autonomous.

Qun et al. [25] used a hybrid optimization approach (using Artificial Bee Colonies (ABC-ACO) and Ant Colony Optimization) to build an energy awareness for load balancing in fog-based VANETs. The Network Simulator 2 (NS2) environment's simulation findings demonstrated that the VANET's energy consumption rose with node count. Additionally, load balancing using the suggested way has improved as the number of jobs has increased. The simulation trials ultimately shown that the suggested hybrid algorithm, ACO-ABC, works better than any other method.

Li et al. [26] have looked at a nature-inspired fuzzy-based resource allocation technique for automotive cloud computing called the Cuckoo Search Algorithm. Landlords may rent these resources, or cars can share them, for a variety of uses, including supplying the hardware required by automobile network services and apps. It is feasible to provide the automobile network's expanding resource needs. The results demonstrated how the suggested approach performs better than other algorithms in terms of makespan, latency, and execution time. A resource allocation method for VANETS that draws inspiration from the banker's algorithm has been introduced by Balzano et al. [27]. By using this approach, we handle cars as processes that make requests and the roads as resources that need to be distributed. We then provide an algorithm to control the distribution of vehicles along the available pathways in order to lessen traffic congestion.

3. Proposed Methodology

In this research methodology to develop and evaluate their proposed hybrid optimization algorithm for resource allocation in multi-cloud environments for VANETs. The first step is to form a network of all connectable vehicles in the VES. Once the network is formed, features are extracted from each vehicle, such as its location, speed, direction of travel, and available resources. The next step is to select a subset of the extracted features that are most relevant to the resource allocation problem. This can be done using a variety of feature selection methods, such as the Walrus Optimization Algorithm and Osprey Optimization Algorithm (WaOA-OOA). Once the features have been selected, a cost-time matrix is computed for each task and vehicle. The cost-time matrix represents the cost and time required to allocate a task to a particular vehicle. The next step is to cluster the vehicles into groups using a variety of clustering algorithms, such as the Farthest First K-Means algorithm. The task allocation problem is then solved using the Drawer Algorithm (DA), a meta-heuristic algorithm that is well-suited for solving complex optimization problems. The performance of the proposed resource allocation algorithm is finally evaluated using a cloud server. The cloud server can be used to simulate the VANET environment and to generate realistic task requests.

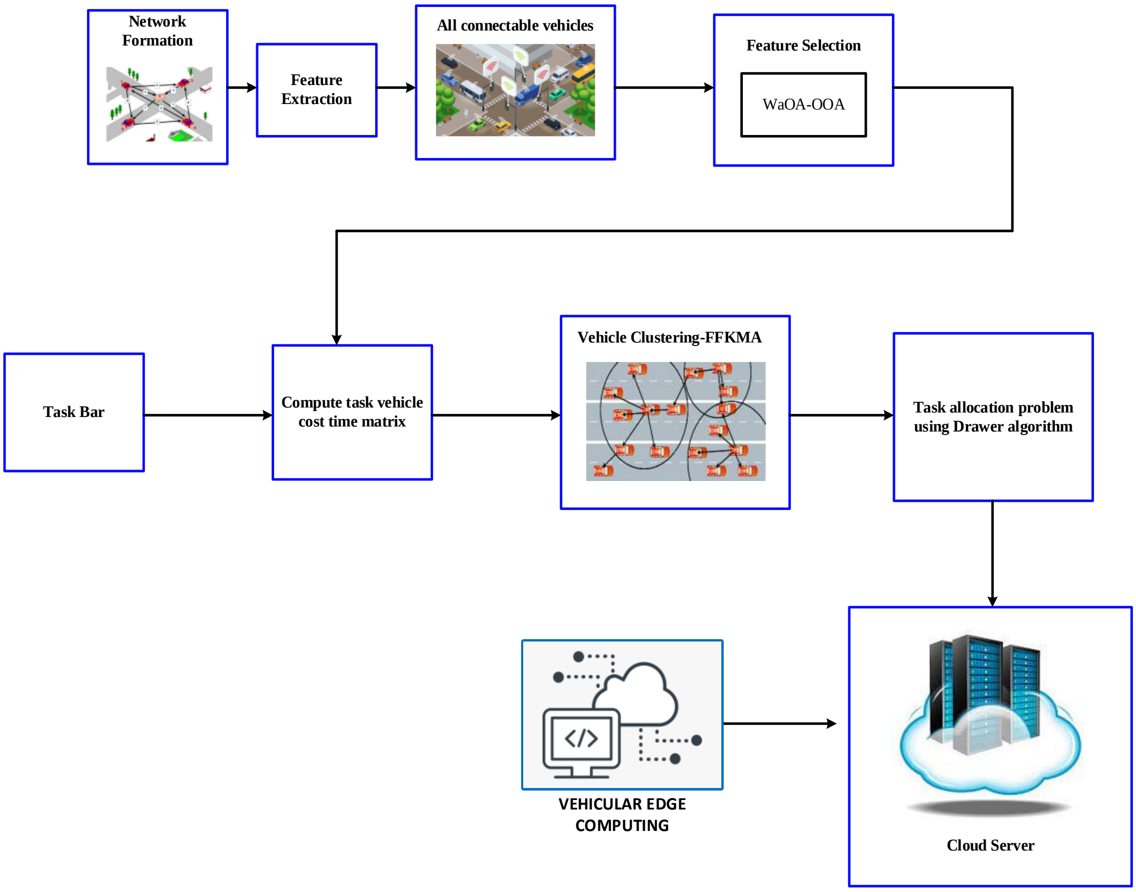


Figure 1 Overview of proposed methodology

3.1. Network Formation

The compute and communication utilities make up the two components of the VEC network's utility.

Communication utility

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

The unit price of transmitting is specified as $\alpha_x(t)$ per bps at time slot t, and is charged by the VEC operator to User Equipments (UEs) u_x for sending calculation tasks to VES. In the meanwhile, the VEC operator leases wireless network backhauls and spectrum inside VES [28]. Therefore, the VEC operator's communication tool for allocating radio resources u_x at time instant t is,

$$\Psi_{Com,x}(t) = \alpha_x(t)\mu_x - r_x\delta_{VES,x}\gamma^n_{VES,x}(t)$$

(1)

The unit price of leasing spectrum from VES to μ_x is defined as $\delta_{VES,x}$ per bps; the communication time costs depend on the size of calculation input data μ_x, data rate of u_x serviced by $\gamma^n_{VES,x}$.

Computation Utility

Next, the usefulness of the VEC operator in giving UEs access to computing services. For the computing work Ω_x during time slot t, the VEC operator charges UEs a unit fee of $\lambda_x(t)$. Meanwhile, the unit price of the compute resource that VEC operator is $\Delta_{VES,x}$ correspondingly. Let $Cpt_{VES,x}$ signify the VES compute resource that is allotted to the mobile μ_x device at time instant t. As a result,

$$\Psi_{Cpt,x}(t) = \lambda_x(t)Q_x - r_x\delta_{VES}\gamma^n_{VES,x}(t)$$

(2)

Thus, at time instant t, the total utility provided by the VEC operator to the UE may be computed as follows:

$$\Psi_x(t) = \Psi_{com,x}(t) + \Psi_{cpt,x}(t) \tag{3}$$

Lastly, we provide the VEC operator's utility function as follows:

$$\Psi_x(t) = \sum_{x\in X} D_x\Psi_x(t)$$

$$= \sum_{x\in X} D_x\left(\Psi_{com,x}(t) + \Psi_{cpt,x}(t)\right)$$

(4)

3.2. Feature Extraction

In the context of resource allocation for multi-cloud environment in vehicular ad-hoc networks (VANETs), feature extraction can be used to identify the most important factors that influence resource allocation decisions.

Vehicle speed: Vehicles that are traveling at high speeds require more resources to maintain communication with the cloud.

Traffic density: When traffic is dense, there is more competition for resources, so it is important to allocate resources carefully.

Bandwidth availability: Bandwidth availability will vary depending on the cloud provider. It is important to allocate resources to cloud providers that have sufficient bandwidth to meet the needs of the applications.

Latency: Latency is the time it takes for a packet of data to travel from one point to another. It is important to allocate resources to cloud providers that have low latency to ensure that applications perform well.

These features are important because they directly impact the ability of vehicles to communicate with the cloud and with each other. Vehicles that are traveling at high speeds or in dense traffic require more resources to maintain communication. Cloud providers with limited bandwidth or high latency may not be able to meet the needs of all vehicles in the network. This can help to improve the efficiency and performance of VANETs by ensuring that resources are allocated to the most important applications and users.

3.3. Feature selection using hybrid Walrus Optimisation Algorithm and Osprey Optimisation Algorithm (WaOA-OOA) in vehicular edge computing

Feature selection is the process of selecting the most important features from a large set of features. This is important because it can help to improve the performance of hybrid Walrus Optimisation Algorithm and Osprey Optimisation Algorithm (WaOA-OOA) models by reducing the number of features that need to be processed and by removing irrelevant or noisy features. The hybrid WaOA-OOA has several benefits for feature selection task in vehicular edge computing (VEC). First, it can help to improve the performance of machine learning models in VEC applications, such as traffic forecasting, anomaly detection, and recommendation systems. Second, it can help to reduce the complexity of WaoA algorithm using OOA model in vehicular edge computing, which can be important for devices with limited resources, such as smartphones and on-board units in vehicles. Lastly, it may aid in improving the hybrid WaoA-OOA models' interpretability in applications involving vehicular edge computing, which might be crucial for comprehending and troubleshooting the models. Walruses make up the searcher members of the WaOA population-based metaheuristic algorithm. Every walrus in WaOA stands for a potential fix for the optimization issue. Consequently, the candidate values for the problem variables are determined by each walrus's location inside the search space. As a result, every walrus is a vector, and the so-called population matrix may be used to numerically represent the walrus population. Walrus populations are randomly established at the start of WaOA implementation [29]. Using (5), this WaOA population matrix is calculated.

$$Q = \begin{bmatrix} Q_1 \\ \vdots \\ Q_i \\ \vdots \\ Q_K \end{bmatrix}_{K \times l}$$

(5)

where K is the number of walruses, l denotes the number of choice variables, Q_i is the ith walrus (candidate solution) and population walruses' denoted by Q .

The primary goal of the suggested hybrid optimized resource allocation model is to reduce vehicle power use while still allowing for the evaluation of vehicle delay. The goal function's approximated values derived from walruses are listed in,

$$OF = Min\left[(V_{PC}),(D)\right]$$

(6)

Where, V_{PC} is the minimization of power consumption and D is the vehicle delay.

Phase 1: Feeding strategy (exploration)

Over sixty different types of marine invertebrates, including shrimp, sea cucumbers, tunicates, soft corals, tube worms, and numerous mollusks, are among the diverse foods that walruses consume. The walruses' search activity results in various scanning regions of the search space, which strengthens the WaOA's exploration capability during the global search. Using equations (7) and (8), the feeding mechanism is used to quantitatively simulate the process by which walruses update their location, with the leader of the group serving as the guide. The initial step in this procedure is to create a new position for Walrus in accordance with (7). If this new location increases the value of the goal function, it takes the place of the prior one; this idea is described in (8).

$$q_{x,y}^{Pn_1} = q_{x,y} + Rand_{x,y} * \left(sw_y - In_{x,y} \cdot q_{x,y} \right)$$
(7)

$$Q_x = \begin{cases} Q_x^{Pn_1}, \ OF_x^{Pn_1} < OF_x \\ \quad Q_x, \quad Else \end{cases}$$
(8)

Here $Rand_{x,y}$ are random numbers from the interval [0, 1], $In_{x,y}$ are integers randomly chosen between 1 and 2, are the best candidate solution, which is thought to be the strongest walrus, and $Q_x^{Pn_1}$ are the newly generated position for the xth walrus based on the first phase, $q_{x,y}^{Pn_1}$ its yth dimension, and $OF_x^{Pn_1}$ its objective function value.

Phase 2: Migration

Due to the air warming in the late summer, walruses migrate to stony beaches or outcrops as part of their normal activity. According to this modeling, every walrus moves to a different (randomly chosen) location in a different part of the search space. Consequently, the suggested new position is first created using (9). Then, in accordance with (10), the new location takes the place of the walrus if it increases the value of the goal function.

$$q_{x,y}^{Pn_2} = \begin{cases} q_{x,y} + Rand_{x,y} * \left(q_{z,y} - In_{x,y} \cdot q_{x,y} \right), OF_z < OF_x \\ \quad q_{x,y} + Rand_{x,y} * \left(q_{z,y} - q_{x,y} \right), Else \end{cases}$$
(9)

$$Q_x = \begin{cases} Q_x^{Pn_2}, \ OF_x^{Pn_2} < OF_x \\ \quad Q_x, \quad Else \end{cases}$$
(10)

Where $q_{x,y}^{Pn_2}$ is its yth dimension, $OF_x^{Pn_2}$ is its objective function value, q_z, $q_z \in \{1, 2, ..., k\}$ and $z \neq x$ is the location of the chosen walrus to move the xth walrus in its direction. $Q_x^{Pn_2}$ are the new produced position for the xth walrus based on the first phase, $q_{z,y}$ its yth dimension, and objective function value is denoted by OF_z . Following that, the fish updating procedure—which is derived in the equation below—is discovered in order to determine the safe location for hunting (11), One of these fish is randomly located by the osprey [30], which then attacks it. A new location for the matching osprey is determined using (12) based on the simulation of the osprey's journey towards the fish. As per equation (13), the osprey's original location is replaced by this new one if it enhances the value of the goal function.

$$w_{i,j}^{O1} = w_{i,j} + \alpha_{i,j} . \left(E_{i,j} - D_{i,j} . w_{i,j} \right)$$
(11)

$$w_{i,j}^{O1} = \begin{cases} w_{i,j}^{O1}, & v_j \le w_{i,j}^{O1} \le u_j; \\ v_j, & w_{i,j}^{O1} < v_j; \\ u_j, & w_{i,j}^{O1} > u_j. \end{cases}$$

(12)

$$W_i = \begin{cases} W_i^{O1}, & R_i^{O1} < R_i; \\ W_i, & else. \end{cases}$$

(13)

Where $\alpha_{i,j}$ are the random numbers in the interval [0, 1] and $D_{i,j}$ the random numbers from the set {1, 2}, W_i^{O1} are the new location of the i^{th} osprey based on the first phase of osprey, $w_{i,j}^{O1}$ are its j^{th} dimension, R_i^{O1} is the objective function value, $E_{i,j}$ is the fish that the i^{th} osprey has chosen.

Phase 3: Escaping and fighting against predators (exploitation)

Killer whales and polar bears are constant threats to walruses. The walruses' position in relation to their current location changes as a result of their escape and defense tactics against these predators. Each walrus is believed to have a neighborhood, and utilizing (14) and (15), it first generates a new position at random inside this neighborhood. Then, in accordance with (16), this new position takes the place of the prior position if the value of the goal function is increased.

$$q_{x,y}^{Pn_3} = q_{x,y} + \left(L_{lb,y}^{n} + \left(L_{ub,y}^{n} - Rand \cdot L_{lb,y}^{n} \right) \right)$$

(14)

$$Local\,bounds: \begin{cases} L_{lb,y}^{t} = \dfrac{lb_y}{t} \\ L_{ub,y}^{t} = \dfrac{ub_y}{t} \end{cases}$$

(15)

$$Q_x = \begin{cases} Q_x^{Pn_3}, \; OF_x^{Pn_3} < OF_x \\ Q_x, \quad Else \end{cases}$$

(16)

In order to mimic local search in the vicinity of the candidate solutions, where $Q_x^{Pn_3}$ is the new produced location for the x^{th} walrus based on the third phase, $q_{x,y}^{Pn_3}$ is its y^{th} dimension, $L_{lb,y}^{t}$ and $L_{ub,y}^{t}$ are local lower and local upper limits permissible for the y^{th} variable, respectively.

Input: Obtained features selection in vehicular edge computing

Output: Selected features ζ_i

Begin

Initialize population, fitness, iteration rr_t and maximum iteration x_{ma}

Compute fitness

Set iteration $rr_t = 1$

While $(rr_t \le x_{ma})$ **do**

Feeding strategy (exploration) of the walruses leads by,

$$q_{x,y}^{Pn_1} = q_{x,y} + Rand_{x,y} * \left(sw_y - In_{x,y} \cdot q_{x,y} \right)$$

Update the new position

If $Q_x^{Pn_1} < Q_x^{Pn_2}$ **{**

$$q_{x,y}^{Pn_2} = \begin{cases} q_{x,y} + Rand_{x,y} * \left(q_{z,y} - In_{x,y} \square q_{x,y}\right), OF_z < OF_x \\ q_{x,y} + Rand_{x,y} * \left(q_{z,y} - q_{x,y}\right), Else \end{cases}$$

} else {

$$Q_x^{Pn_2} < Q_x^{Pn_3}$$

$$q_{x,y}^{Pn_3} = q_{x,y} + \left(L_{lb,y}^n + \left(L_{ub,y}^n - Rand \square L_{lb,y}^n\right)\right)$$

} end if

Obtain the safe position for eat prey using, $w_{i,j}^{O1} = w_{i,j} + \alpha_{i,j} . \left(E_{i,j} - D_{i,j} . w_{i,j}\right)$

Calculate fitness

Set $rr_t = rr_t + 1$

End while

Return Cluster head

End

3.4. Task Computation Model

We consider the request to send/clear to send (RTS/CTS) approach in order to minimize collisions [31]. After the time interval SIFS, UAV y may receive the RTS frame and respond with a CTS frame if the wireless channel is idle. Following the time interval SIFS, vehicle x gets the CTS frame and begins transmitting the work. After obtaining the job, UAV y finally replies to vehicle x with an ACK frame. Thus, we establish the duration of the successful transmission job between vehicle x and UAV y as, $\beta_{x,y}^c$ in a way that

$$\beta_{x,y}^c = \beta_{x,y}^t + h_{\Re_x} + 4\beta_{x,y}^p + 3\beta^{SIFS} + \beta^{ACK} + \beta^{DIFS} + \beta^{RTS} + \beta^{CTS}$$

(17)

Where $\beta_{x,y}^p$ denotes the propagation delay and $h_{\Re_x}$ reflects the task $\Re_x$ overhead of the packet header. $\beta^{SIFS}, \beta^{ACK}, \beta^{DIFS}, \beta^{RTS}$ and β^{CTS} indicate, in that order, the SIFS frame interval, ACK frame interval, DIFS frame interval, RTS frame interval, and CTS frame interval. Furthermore, $\beta_{x,y}^t$ denotes the transmission delay, which is expressed as,

$$\beta_{x,y}^t \approx \frac{\delta_{x,y} D_x}{BW_y \log_2\left(1 + \frac{Pw_x^{\Re_x} \cdot g_{x,y}}{N_p}\right)}$$

(18)

Where $Pw_x^{\Re_x}$ stands for the vehicle x's transmission power, BW_y the UAV j's bandwidth, the N_p receiver input's noise power, and $g_{x,y}$ the channel gain between the UAV y and the vehicle x.

The wireless channel collision period may be represented as $\beta_{x,y}^c$ follows:

$$\beta_{x,y}^c = \beta^{RTS} + \beta^{DIFS} + \beta_{x,y}^p$$

(19)

So, using this information, the normalized throughput $h_{x,y}$ may be computed as

$$h_{x,y} = \frac{p^c_{x,y}\delta_{x,y}D_x}{p^f_{x,y}\Delta + p^c_{x,y}\beta^c_{x,y}}$$
(20)

Δ indicates how long a timeslot lasts.

In order to guarantee that the work may be finished on time, the limited computational resources of the UAV y must be distributed as equally across the S vehicles as feasible. Let T^F_y indicate the overall computing frequency of UAV y and $T^F_{x,y}$ the computing frequency that UAV y has allocated to vehicle x, i.e $T_F = \{F_{1,y}, F_{2,y}, ..., F_{S,y}\}$. We've got $\sum_{x=1}^{S} n_{x,y} T^F_{x,y} \le T^F_y$. The VEC server's $\beta^{Edge}_{x,y}$ compute time on UAV y may be written as

$$\beta^{Edge}_{x,y} = \frac{\delta_{x,y} c_x}{T^F_{x,y}} \tag{21}$$

In this instance, we define the work offloading issue in a vehicular cloud as an optimization issue (22). The system utilizes an offloading decision matrix Ω that minimizes the overall task completion time as the objective function while meeting the constraints provided by $c_1 - c_4$, given the number of subtasks M' and the number of vehicles n.

$$M \text{ in} \begin{cases} Max\left\{\sum_{x=1}^{M'} f_{xy}.D_{xy}\right\}, y = 1,2,3,....,i \\ c_1 : f_{x1} + f_{x2} + ... + f_{xi} = 1, x = 1,2,...,M' \\ c_2 : f_{xy} = 0 \ or \ f_{xy} = 1 \\ c_3 : Max\left\{\sum_{x=1}^{M'_n} f_{xy}.D_{xy}\right\} \le P_n, y = 1,2....,i \\ c_4 : \sum_{x=1}^{M'} f_{xy}.D_{xy} \le rt(y),\ y = 1,2,...,i \end{cases} \tag{22}$$

Where M'_n is the total number of subtasks that this task has produced. The time remaining for the vehicle to remain in the vehicular cloud is called the maximum tolerated delay (P;), $rt(y)$ and it must be met by the completion of each job n. There c_1 is a constraint that only one vehicle may be assigned to each subtask. shows that or not when subtask x is transferred to vehicle j. c_2 shows that $f_{xy} = 1$ each work must be finished within its delay budget unless subtask x is transferred to vehicle j or $f_{xy} = 0$. c_3 demands that every work be finished within the allotted delay budget. c_4 mandates that every vehicle do all of the work assignments that have been delegated to it during the remaining time it has in the vehicular cloud (i.e., before the car departs the present vehicular cloud).

3.5. Vehicle clustering for vehicular edge computing using Farthest First K-Means algorithm

For clustering the obtained data from different locations vehicle node density (ND), speed, vehicle node distance (NDS), and residual energy (RE), bandwidth and storage this research methodology uses the Farthest First K-Means algorithm (FFKMA). In this algorithm the initial centroid point is selected randomly and the farthest point from the initialized centroid is considered as the next centroid point. But directly selection of the centroid attains poor clustering performance. So that the information gain is calculated for the input features based on the information gain the centroids are chosen. Farthest-First operates in two steps of process:

- Information gains the centroids selection
- Cluster assignment

The first stage selects a random data location to serve as the first cluster center. The data point that is farthest from the initial center is determined to be the next center during the cluster assignment step. Similar criteria are used to choose the next centers: they must be the farthest from the group of centers that were first selected. The algorithm ends when a desired number of X centroids has been selected, allocating all subsequent data points to the cluster represented by the closest centroid. The cars close to the CH serve as an example of the second node density criterion because of their dispersion in the region [32]. Because of this, each vehicle has a unique number of neighbor nodes. When vehicles choose a vehicle as the CH and there aren't many other vehicles in their transmission range, they can't transmit directly to the CH and have to use intermediary nodes, which increases transmission costs.

As a result, FFKMA could organize a group of data points with only one pass. The Farthest-First clustering technique differs from K-means clustering in that all centroids are real data points rather than the geometric centers of clusters because no normal attribute positions are computed to update the centroids. This technique, which is a 2-approximation to determining the optimal set of centroids with regard to inter-cluster distance—also known as the -centers problem—achieves high performance in X centroid selection despite beginning with a random selection and only requiring one pass. Stated differently, the separation between the centroids chosen by the FFKMA algorithm falls within a factor of two of the ideal distance $[W_i X_i]$.

Input: Obtained cluster sets
Output: Selected vehicles

Begin

Initialize population, fitness, iteration I_t and maximum iteration MI_t

Compute fitness

Set iteration $I_t = 1$

While $(I_t \leq MI_t)$ **do**

Finding a third point which is the farthest point from the first two existing points $[W_i X_i]$

Henceforth $i = 1,2,3....,X$

Calculate fitness

Set $I_t = I_t + 1$

$P_1, P_2......, P_X$ are points or objects of dataset belongs to cluster

End while

Return k-means

End

Thus, the K-means the centroids are chosen are given as follows,

$$Min\{Max[Dis(P_i, P_1), Dis(P_i, P_2).....]\} \tag{23}$$

Where, Dis defines the distance cluster sets and P_i defines the i-number of points from the selected cluster centroids.

3.6. Minimizing Task resource allocation problem in Vehicular edge computing using drawer algorithm (DA)

Vehicles are grouped for each task, and then the best resource assignment plan is chosen to reduce the total task completion time. Determines the completion time of each subtask on its candidate cars before allocating jobs to vehicles, creating a cost-delay matrix Ω in the process. Each entry in the matrix Ω, which consists of M' rows and n columns, Ω_{xy} reflects the time it took for subtask-x on the vehicle v_y to be completed. Ω_{xy} is determined using (25); v_y if is determined to be a potential vehicle for subtask-x, Ω_{xy} is set to infinity.

$$\Omega_{xy} = \begin{cases} \dfrac{cc_x \cdot s_d}{c_{ca}^{y}}, & y = z \\ \dfrac{cc_x \cdot s_d}{c_{ca}^{y}} + \dfrac{s_d}{R}, & y \neq z \end{cases}$$

(24)

Where s_d is the data amount of a subtask, is the transmission rate between two cars, cc_x is the computational complexity of the x-th subtask, and c_{ca}^{y} is the computing capability of the vehicle v_y, $y = z$ which represents the scenario where this subtask is processed locally (i.e., without being offloaded). Assigning M' separate subtasks to n cars is the aim of this stage, which aims to reduce the overall task completion time. The formulation of this issue is a 0-1 integer linear programming problem, which has been shown to be NP-hard by (20). The challenge is figuring out how to divide up the few VEC resources among the many cars in a manner that optimizes the system's overall performance. A simple but efficient method for resolving the resource allocation issue in VEC is the drawer algorithm (DA). Each drawer in the DA represents a distinct collection of resources, and cars are iteratively assigned to them. For every car, the program first creates a drawer [33]. Subsequently, it allocates cars to drawers in an iterative manner so as to optimize the use of resources in every drawer.

Procedure of DA algorithm

Step 1: Initialization

At the beginning of the drawer implementation, equation (25) is used to initialize the drawer for every vehicle point in the search area at random.

$$R = \begin{bmatrix} \vec{R}_1 \\ \vdots \\ \vec{R}_x \\ \vdots \\ \vec{R}_K \end{bmatrix}_{K \times L}$$

(25)

Step 2: Fitness function calculation

The primary goal of this article is to provide methods for reducing the time it takes to complete an activity.

$$OF = Min\left[D_x^{TC} \right]$$

(26)

Where D_x^{TC} denotes the reduction of delay units that propels the completion of job x.

Step 3: Update function

Here, a well-chosen mix of drawers holding variable information will be used to update the population matrix in the DA. More precisely, the DA counts on a commode having the same number of drawers as the variables in the optimization problem. The recommended values for the associated variables are different in each of the commode's drawers. The following equations may be used to mathematically represent the commode and drawers:

$$dr = \begin{bmatrix} \vec{d}r_1 \\ \vdots \\ \vec{d}r_x \\ \vdots \\ \vec{d}r_K \end{bmatrix}_{L\times 1} \tag{27}$$

$$dr_n(i) = \left[\left(1-\frac{i}{I}\right)*n\right], i = 1,\dots\dots, I$$

(28)

$$\vec{d}r_y = \left(f_{Rand}(n), y \middle| z = 1,2,\dots\dots, dr_n(i)\right), y = 1,\dots\dots, m$$

(29)

where I is the overall number of iterations, $dr_n(i)$ is the number of drawers in the ith iteration, dr is the drawer matrix, $\vec{d}r_y$ is the vector of values in the yth drawer, $y = 1,\dots\dots, m$ represents the standard mathematical ceiling function, and $f_{Rand}(n), y$ is the corresponding element of the yth column of the $Rand(n)$ row. $Rand(n)$ is the random function, which uniformly generates a random number from the set $\{1,\dots\dots, n\}$.

Step 4 Update each member of the population

A random combination of drawers is formed in such a way that each drawer has precisely one value, which is taken into consideration for determining the value of a problem variable. The chosen values from the drawers are then combined to create a random combination that directs the population member. The method used to create this arbitrary combination is described as,

$$\vec{\alpha}_x = \left\{ g_{y,Rand(dr_n(i))} \middle| y = 1,2,\dots, m \right\}, x = 1,2,\dots, n$$

(30)

Where $g_{y,Rand(dr_n(i))}$ is the corresponding element of the yth row of the $Rand(dr_n(i))$ th column of the matrix dr, $\alpha_{x,y}$ is its yth dimension, and $\vec{\alpha}_x$ is the random combination to guide the xth population member; $Rand(dr_n(i))$ is a function that creates a random integer from the set $\{1,2,\dots\dots,(dr_n(i))\}$.

Step 5: Calculate a new status of population member

Each member of the population is updated in the search space using the expressions provided above, after the random composition has been established.

$$f_{x,y}^{*} = \begin{cases} f_{x,y} + Rand(0,1)\square\left(\alpha_{x,y} - Rand(2)\bullet f_{x,y}\right), OF_x^c < OF_x \\ f_{x,y} + Rand(0,1)*\left(f_{x,y} - \alpha_{x,y}\right) \qquad else \end{cases}$$

(31)

$$F_x = \begin{cases} F_x^*, F_x^* \le F_x \\ F_x, \; else \end{cases}$$

(32)

Where F_x^* are the objective function value, f_x^* the new status of the xth suggested solution, $f_{x,y}^*$ its yth dimension, and OF_x^c the objective function of the random combination to direct the xth population member. Lastly, the DA is used to efficiently distribute tasks among cloud servers, minimize resource consumption, and guarantee that tasks are finished within the allotted timeframes. Constantly monitoring and optimizing cloud server resources is necessary to maintain optimal resource allocation and adjust to changing circumstances.

4. Results and Discussion

In order to illustrate the simulation results of the best resource allocation in the VANET's VEC, provide the methodology's simulation results in this part. Using the permutation-based model, the resource allocation of cluster-based algorithms and WaOA-OOA-based algorithms is provided, respectively. All of the previously listed situations are explicitly simulated and the programmability of edge nodes is also investigated with the aid of the MATLAB toolbox. The design is put to the test against other approaches that are already in use, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, and Coati Optimization Algorithm (COA), in order to show that successful, it is.

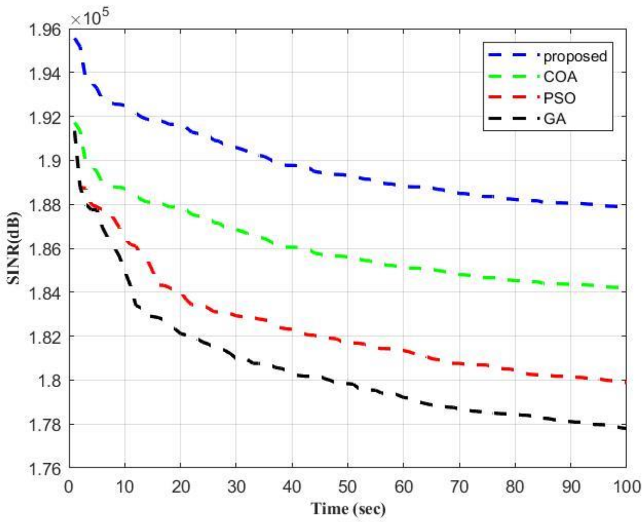
4.1. Performance Matrices

Here, the suggested research methodology's performance is compared to that of the current research techniques in terms of throughput, delay, energy conservation, execution time, SNR, and energy efficiency.

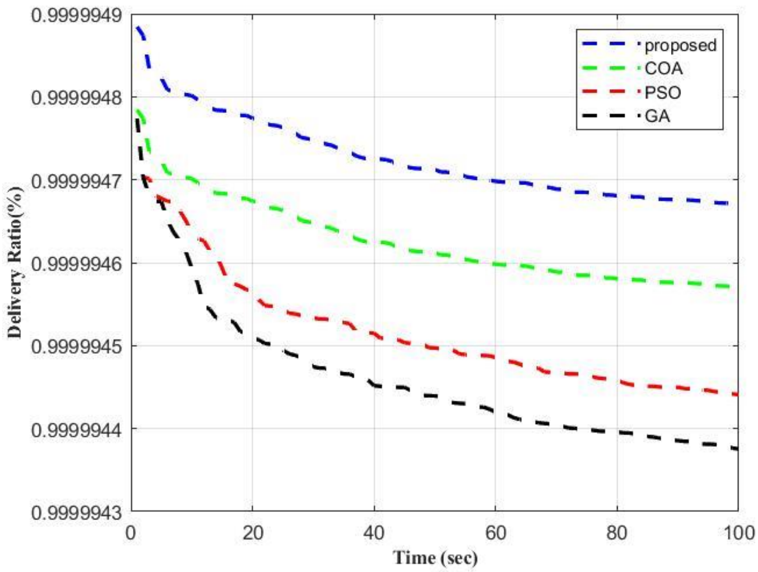
Packet Delivery Ratio (PDR): This is the proportion of total packets delivered to total packets received. The value falls between 0 and 1. A higher number is preferred.

Average delay: It is the mean amount of time it takes for a packet to travel from sender to recipient. It is measured in milliseconds. A lower value is preferred.

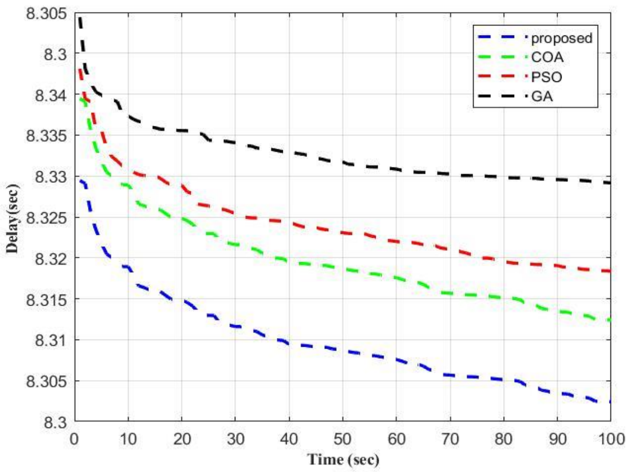
Throughput: It is the amount of data sent each minute. Mbps is the unit of measurement. Optimal value should be the goal.



(a)



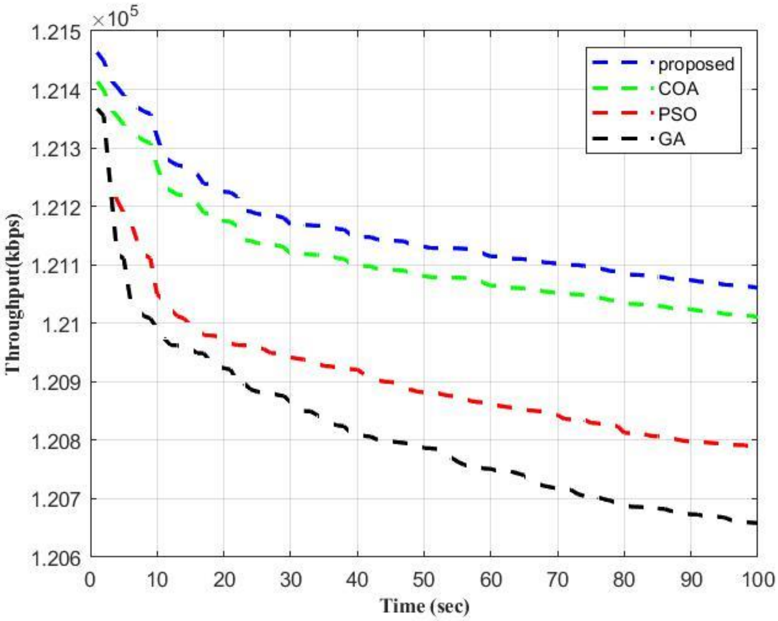
(b)



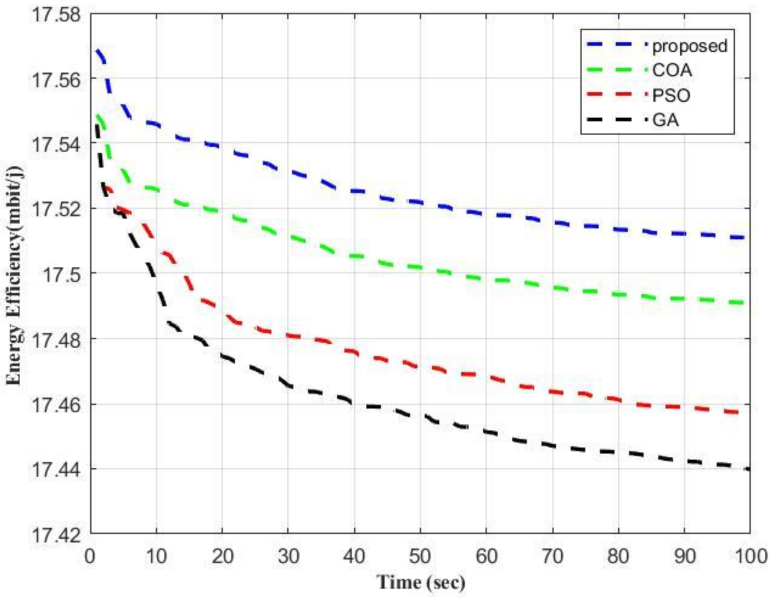
(c)

Figure 2 Performance analysis of (a) SNR (b) Delivery Ratio and (c) Average delay

In terms of (a) SNR, (b) Delivery ratio and average delay are shown in Figure 2 compares the suggested research approach with the research methodologies that are already in use, such as COA, PSO, and GA. A study of the SNR output in the suggested and current VEC approaches is shown in Figure 2(a). In this regard, the suggested method advances the idea of VEC centroid localization by giving weights to the coordinates of each surrounding vehicle based on SNR values and distances. When compared to the suggested research technique, the COA, PSO, and GA achieve lower values, but the SNR value of the suggested method is over $1.88*10^5$. Due to the fact that the suggested study approach makes use of the VEC's superior optimum resource locating function. Figure 2(b) displays the delivery ratio graph with a variable number of times. When compared to other current approaches, the WaOA-OOA method that is being presented exhibits reduced latency. The reason for this is because the RSUs detect the channels and utilize the perceived data to train themselves. RSUs are educated in a way that facilitates their choice of a channel where the likelihood of a VEC arrival is low. The RSUs' ability to sense channels saves time as well. All approaches show an increase in delay as the number of vehicles grows. It is a result of the network's increasing congestion. The delivery ratio values of the suggested approach vary by 0.99999468% from the values of the current methods, which are COA at 0.9999948%, PSO at 0.99999445%, and GA at 0.99999438%. In figure 2(c) shows that the comparison analysis of average delay has illustrated. The proposed approach is contrasted with traditional approaches like PSO and GA. The delay of the proposed method is 8.3022 sec. The COA, PSO and GA both have delay values of 8.3125 sec, 8.319 sec and 8.329, at 100 sec respectively.



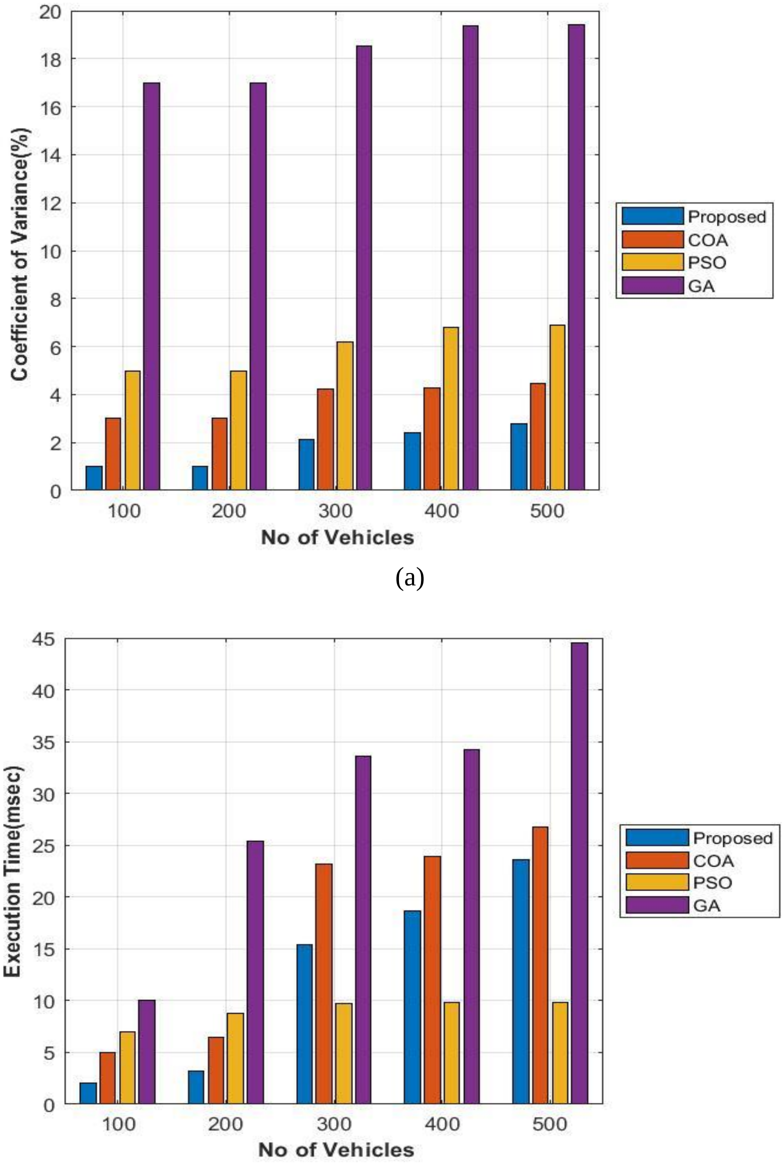
(a)



(b)

Figure 3 Performance analysis of (a) Throughput and (b) Energy Efficiency

The suggested technique's throughput is shown in Figure 3(a). This is accomplished using the suggested method at $1.215*10^5$. Utilizing the traditional methods of COA, PSO, and GA, $1.21*10^5$, $1.208*10^5$, and $1.2065*10^5$are obtained. The study indicates that the suggested approach achieves efficient throughput earnings. The energy efficiency study of suggested and current technologies is shown in figure 3(b). With an energy efficiency of 17.51, the suggested approach is the most energy efficient, followed by the COA techniques (17.49), PSO method (17.46), and GA method (17.44). The suggested model's performance throughput and energy efficiency are shown in the findings.

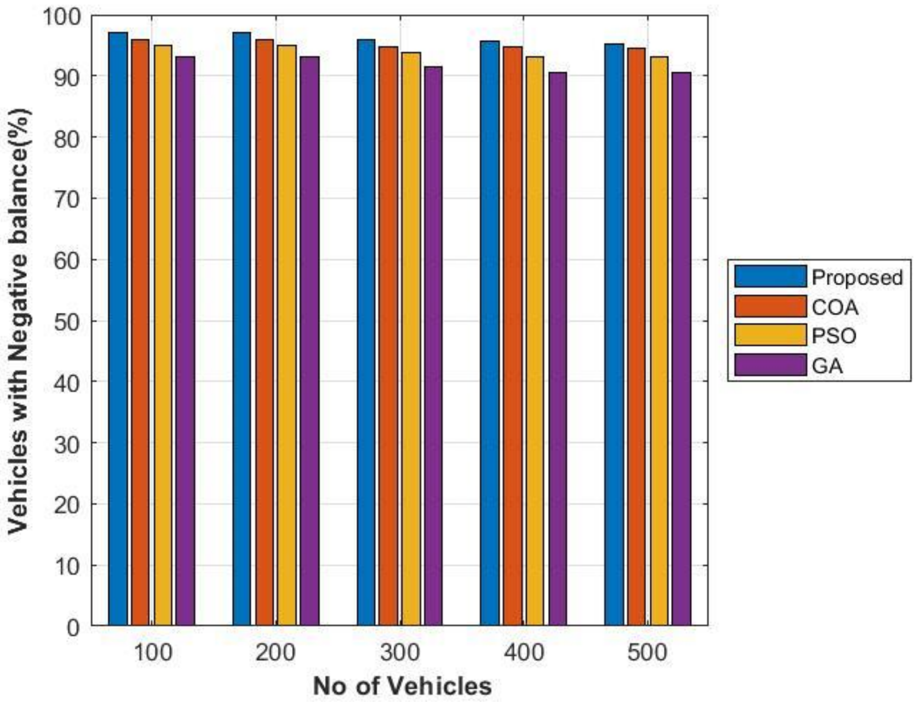


(a)

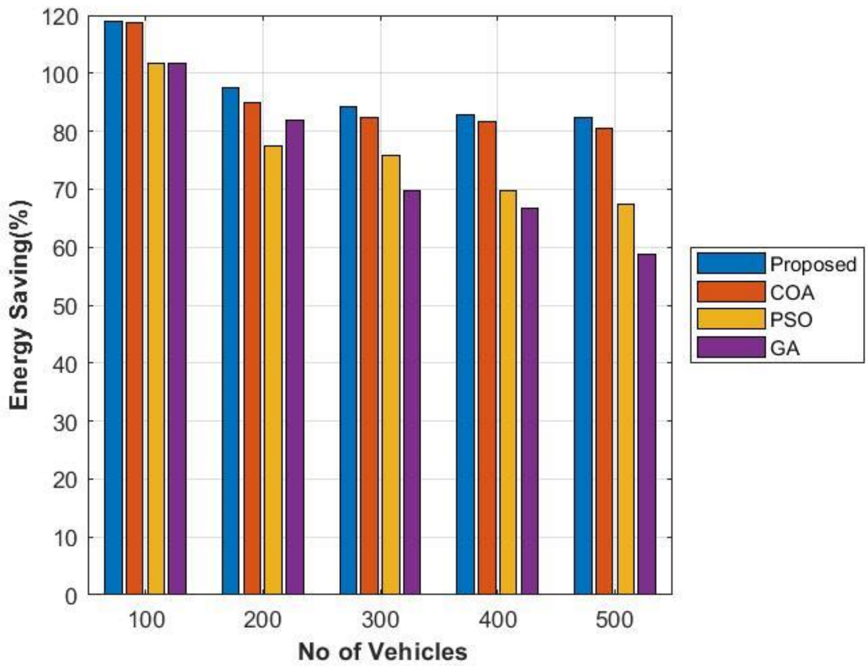
(b)

Figure 4 Outcome of (a) Coefficient of variance and (b) Execution time

The comparative study between the execution time and the coefficient of variation is shown in figure 4. The analysis and simulation models for the coefficient of variance are shown in Figure 4(a). As a consequence, the arrival rate is Poisson and the difference in the findings was quite small. There is, thus, a clear distinction between the simulation and analytical outcomes. We took into consideration the limited system capacity of vehicles with the ranges between 100 veh/s, 200 veh/s, 300 veh/s, 400 veh/s, and 500 veh/s with the variation range from 0 to 20% in VANETs, the highly dynamic vehicular edge computing of vehicular flow. The suggested approach the 100 veh/s at 17%, 200 veh/s at 17%, 300 veh/s at 18.5%, 400 veh/s at 19%, and 500 veh/s at 19% are the coefficients of variation. The suggested method has a higher coefficient of variance analysis than the other three. As the number of cars increases, both the suggested and current systems show polynomially increasing execution durations, as seen in Figure 4(b). Since these systems' execution times consistently remain less than 27% of the energy-manager period, it is acceptable.



(a)



(b)

Figure 5 Performance analysis of (a) Vehicles with negative balance and (b) Energy Saving

In Figure 5(a), the cars with negative balance output are shown. When compared to the current approaches (red, yellow, and purple lines), it is evident that the suggested method (blue line) consistently produces a maximum 95% of cars with negative balance across varied numbers of vehicles in the VANET. This suggests that there is a greater chance of resource depletion for cars using the suggested resource distribution strategy. The comparative study of energy saving is given in Figure 5(b). Here is a percentage (%) representation of the energy savings achieved by each approach. This axis shows how much energy is saved using the suggested approach in comparison to the three other approaches currently in use. The blue result (119%, 95%, 90%, 89%, and 89%) shows how much energy is saved using the suggested strategy as the number of cars rises. In a similar vein, the energy savings of the current COA, PSO, and GA methods are dependent on the quantity of cars. This comparison shows how much better the suggested strategy performs in terms of energy conservation.

5. Conclusion

This study introduces a cutting-edge hybrid optimized resource allocation model tailored to the unique challenges related to Vehicular Edge Computing (VEC). This innovative model takes into account a multitude of factors, such as resource availability, geographical distances, network bandwidth, and task-specific requirements, to streamline resource allocation and enhance the efficiency of task execution within VEC. By using the combination of the Walrus Optimization Algorithm and the Osprey Optimization Algorithm (WaOA-OOA) the design is refined which promotes efficient allocation. The proposed model's core lies in the computation of a task-vehicle cost-time matrix, integrating resource capabilities and task demands. This guides the vehicle clustering process, employing the Farthest First K-Means algorithm to create clusters of vehicles with similar resource profiles. Importantly, the model seamlessly integrates with cloud servers, securely storing allocation results for efficient task execution. Simulation results validated using MATLAB platform with this the effectiveness of this approach and comparative analysis against existing algorithms that includes the COA, PSO and GA underscores the superiority of the proposed model in terms of throughput, Signal-to-Noise Ratio (SNR), delivery ratio, average delay, and energy savings. This research represents a significant advancement in resource allocation for both cloud and edge computing in VEC, offering a comprehensive solution that optimizes resource utilization and enhances the state of resource management in VEC environments.

References

[1] Rana, S., Mishra, D., Lal, C. and Conti, M., 2023. Authenticated Message-Exchange Protocol for Fog-Assisted Vehicular Cloud Computing. Wireless Personal Communications, pp.1-18.

[2] Bi, C., Li, J., Feng, Q., Lin, C.C. and Su, W.C., 2023. Optimal deployment of vehicular cloud computing systems with remote microclouds. Wireless Networks, pp.1-13.

[3] Gong, M., Yoo, Y. and Ahn, S., 2023. Vehicular Cloud Forming and Task Scheduling for Energy-Efficient Cooperative Computing. *IEEE Access*, *11*, pp.3858-3871.

[4] Rizvi, S.R., Zehra, S., Olariu, S. and El-Tawab, S., 2023, March. ADAM: An Auction-based Datacenter Management in Vehicular Cloud. In 2023 IEEE International Conference on Smart Mobility (SM) (pp. 138-143). IEEE.

[5] Tran, C. and Mehmet-Ali, M., 2023, February. Analysis of Job Completion Time in Vehicular Cloud Under Concurrent Task Execution. In 2023 International Conference on Computing, Networking and Communications (ICNC) (pp. 101-105). IEEE.

[6] Rajput, N.S., Singh, U., Dua, A., Kumar, N., Rodrigues, J.J., Sisodia, S., Elhoseny, M. and Lakys, Y., 2023. Amalgamating Vehicular Networks with Vehicular Clouds, AI, and Big Data for Next-Generation ITS Services. IEEE Transactions on Intelligent Transportation Systems.

[7] Rajput, N.S., Banerjee, R., Sanghi, D. and Kalyansundaram, C., 2023. Evolution of cooperation in vehicular cloud assisted networks for ITS services: A hunt game-based approach. Future Generation Computer Systems, 146, pp.62-77.

[8] Balen, J., Matijašević, M., Pejčić, L., Kundid, J. and Azeem, M.M., 2023, April. Integrating Vehicular Clouds For Supporting Smart Transportation Infrastructure. In 2023 4th International Conference on Smart Grid Metrology (SMAGRIMET) (pp. 1-6). IEEE.

[9] Ribeiro Jr, A., da Costa, J.B., Rocha Filho, G.P., Villas, L.A., Guidoni, D.L., Sampaio, S. and Meneguette, R.I., 2023. HARMONIC: Shapley values in market games for resource allocation in vehicular clouds. Ad Hoc Networks, p.103224.

[10] Yan, G., 2023, May. BlockChain Based Data Security Mechanism for Vehicular Networks. In 2023 6th International Conference on Electronics Technology (ICET) (pp. 658-663). IEEE.

[11] Xiao, H., Zhao, J., Feng, J., Liu, L., Pei, Q. and Shi, W., 2023. Joint Optimization of Security Strength and Resource Allocation for Computation Offloading in Vehicular Edge Computing. IEEE Transactions on Wireless Communications.

[12] Saxena, U.R. and Alam, T., 2023. Recommendation-based trust computation and rating prediction model for security enhancement in cloud computing systems. Service Oriented Computing and Applications, pp.1-19.

[13] Fashakh, A. and İbrahim, A.A., 2023, June. A Proposed Secure and Efficient Big Data (Hadoop) Security Mechanism using Encryption Algorithm. In 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-6). IEEE.

[14] Al-Shabi, M., 2023. An Efficient Delay Aware Emergency Message Dissemination and Data Retrieval in Secure VANET-Cloud Environment. Wireless Personal Communications, pp.1-36.

[15] Wang, D., Yi, Y., Yan, S., Wan, N. and Zhao, J., 2023. A node trust evaluation method of vehicle-road-cloud collaborative system based on federated learning. Ad Hoc Networks, 138, p.103013.

[16] Alsumair, L.K.A. and Aldhlan, K.A., 2023, February. Enhancing Data Security in the Cloud Platform by Encrypting the Used Key Dynamically. In 2023 International Conference on Smart Computing and Application (ICSCA) (pp. 1-5). IEEE.

[17] Reddy, G.M., Reddy, M.V., Dinesh, V., Karthikeyan, C. and Kumar, P.A., 2023, March. Smart Vehicle Management based on Vehicular Cloud. In 2023 Second International Conference on Electronics and Renewable Systems (ICEARS) (pp. 850-854). IEEE.

[18] Ju, Y., Chen, Y., Cao, Z., Liu, L., Pei, Q., Xiao, M., Ota, K., Dong, M. and Leung, V.C., 2023. Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*.

[19] Parthasarathi, P., Sangeethaa, S.N. and Nivedha, S., 2023. Secure Group Communication Among IoT Components in Smart Cities. In Recent Advances in Energy Systems, Power and Related Smart Technologies: Concepts and Innovative Implementations for a Sustainable Economic Growth in Developing Countries (pp. 513-532). Cham: Springer Nature Switzerland.

[20] Mohammed, M.A., Lakhan, A., Abdulkareem, K.H., Zebari, D.A., Nedoma, J., Martinek, R., Kadry, S. and Garcia-Zapirain, B., 2023. Homomorphic federated learning schemes enabled pedestrian and vehicle detection system. Internet of Things, 23, p.100903.

[21] Quessada, Matheus S., et al. "A bat bio-inspired mechanism for resource allocation in vehicular clouds." 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, 2021.

[22] He, Yixin, et al. "Joint task offloading, resource allocation, and security assurance for mobile edge computing-enabled UAV-assisted VANETs." Remote Sensing 13.8 (2021): 1547.

[23] Thanedar, Md Asif, and Sanjaya Kumar Panda. "A dynamic resource management algorithm for maximizing service capability in fog-empowered vehicular ad-hoc networks." Peer-to-Peer Networking and Applications (2023): 1-15.

[24] Taha, Mohammad Bany, et al. "A multi-objective approach based on differential evolution and deep learning algorithms for VANETs." IEEE Transactions on Vehicular Technology 72.3 (2022): 3035-3050.

[25] Qun, Ren, and Seyedeh Maryam Arefzadeh. "A new energy-aware method for load balance managing in the fog-based vehicular ad hoc networks (VANET) using a hybrid optimization algorithm." IET Communications 15.13 (2021): 1665-1676.

[26] Li, Can, Xiaode Zuo, and Amin Salih Mohammed. "A new fuzzy-based method for energy-aware resource allocation in vehicular cloud computing using a nature-inspired algorithm." Sustainable Computing: Informatics and Systems 36 (2022): 100806.

[27] Balzano, Walter, et al. "A Resource Allocation Technique for VANETs Inspired to the Banker's Algorithm." International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. Cham: Springer International Publishing, 2022.

[28] Cha, Narisu, et al. "Virtual edge: Exploring computation offloading in collaborative vehicular edge computing." IEEE Access 9 (2021): 37739-37751.

[29] Trojovský, Pavel, and Mohammad Dehghani. "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior." Scientific Reports 13.1 (2023): 8775.

[30] Dehghani, Mohammad, and Pavel Trojovský. "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems." Frontiers in Mechanical Engineering 8 (2023): 1126450.
[31] Liu, Lei, et al. "Asynchronous Deep Reinforcement Learning for Collaborative Task Computing and On-Demand Resource Allocation in Vehicular Edge Computing." IEEE Transactions on Intelligent Transportation Systems (2023).
[32] Kaliyaperuma, Uma Maheswari, Mary Saira Bhanu Somasundaram, and Nickolas Savarimuthu. "Partitioning-Based Data Sharing Approach for Data Integrity Verification in Distributed Fog Computing." International Journal of Engineering & Technology Innovation 13.2 (2023).
[33] Trojovská, Eva, Mohammad Dehghani, and Víctor Leiva. "Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering." Biomimetics 8.2 (2023): 239.