

Doi:10.32604/cmc.2025.065430

## ARTICLE



Tech Science Press

# A Multi-Objective Joint Task Offloading Scheme for Vehicular Edge Computing

# Yiwei Zhang, Xin Cui<sup>\*</sup> and Qinghui Zhao

School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China \*Corresponding Author: Xin Cui. Email: cxsd2007@163.com Received: 12 March 2025; Accepted: 24 April 2025; Published: 03 July 2025

ABSTRACT: The rapid advance of Connected-Automated Vehicles (CAVs) has led to the emergence of diverse delaysensitive and energy-constrained vehicular applications. Given the high dynamics of vehicular networks, unmanned aerial vehicles-assisted mobile edge computing (UAV-MEC) has gained attention in providing computing resources to vehicles and optimizing system costs. We model the computing offloading problem as a multi-objective optimization challenge aimed at minimizing both task processing delay and energy consumption. We propose a three-stage hybrid offloading scheme called Dynamic Vehicle Clustering Game-based Multi-objective Whale Optimization Algorithm (DVCG-MWOA) to address this problem. A novel dynamic clustering algorithm is designed based on vehicle mobility and task offloading efficiency requirements, where each UAV independently serves as the cluster head for a vehicle cluster and adjusts its position at the end of each time slot in response to vehicle movement. Within each UAV-led cluster, cooperative game theory is applied to allocate computing resources while respecting delay constraints, ensuring efficient resource utilization. To enhance offloading efficiency, we improve the multi-objective whale optimization algorithm (MOWOA), resulting in the MWOA. This enhanced algorithm determines the optimal allocation of pending tasks to different edge computing devices and the resource utilization ratio of each device, ultimately achieving a Pareto-optimal solution set for delay and energy consumption. Experimental results demonstrate that the proposed joint offloading scheme significantly reduces both delay and energy consumption compared to existing approaches, offering superior performance for vehicular networks.

**KEYWORDS:** Vehicular edge computing; cooperative game theory; multi-objective optimization; computation offloading

# **1** Introduction

The development of Intelligent Transportation Systems (ITS) and CAVs has brought significant improvements to modern transportation [1]. As a result, numerous compute-intensive and delay-sensitive vehicular applications have emerged in Internet of Vehicles (IoV) scenarios, including navigation, communication services, entertainment, and autonomous driving, etc. [2,3].

Vehicular edge computing (VEC) can coordinate with roadside base stations, edge devices, and cloud servers to distribute the content and functions of centralized networks to the edge, shortening the transmission path between vehicles and servers, thereby enhancing task computing efficiency [4]. In recent years, UAVs have been increasingly used in various fields, taking advantage of their flexibility and ability to operate without geographical limitations. And UAVs can provide edge computing capabilities to ground users [5]. By integrating them into vehicular edge computing, they can effectively reduce the computational load on vehicles, thereby enhancing the overall performance and efficiency of the vehicular network system. Therefore, the UAV-MEC framework has become a research hotspot [6].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In real traffic conditions, vehicles experience high dynamics, coexistence of various services, and uneven distribution of traffic flow in both temporal and spatial domains [7]. IoV applications impose stringent latency requirements and frequently depend on the stability of IoV communications [8].

Recent vehicular edge computing advancements strategically reconcile latency-energy tradeoffs through intelligent task offloading, dynamic resource orchestration, and game-theoretic coordination [9,10]. For task offloading schemes, recent advances employ a hierarchical deep reinforcement learning framework named Sequential Quadratic Programming-based Dueling Double Deep Q Networks reduces system costs and task loss rates [11], Proximal Policy Optimization (PPO)-based method decreases total delay and dropped task ratios [12], a Hybrid Average Reward Proximal Policy Optimization algorithm (hybrid-ARPPO) that combines optimized offloading decisions, transmission rates, and edge server allocation without the discount factor [13], Double Q-learning improving user service quality [14], Digital Twin-assisted Predictive Adaptive Intelligent Offloading scheme to reduce computation latency [15], Exploration and Exploitation Assisted Contract Theory (EEACT) scheme optimal offloading strategies by minimizing task processing costs [16] and a Mobile-Aware Computation Efficiency Task Offloading and Resource Allocation scheme to optimize transmission efficiency [17]. Resource-saving strategies focus on balancing Quality of Service (QoS) and energy consumption through cooperative models [18], PPO-based to balance the long-term cost between minimizing task delay and energy consumption [19], semi-distributed algorithm to solve the task offloading and service caching problem [20], based on Non-Orthogonal Multiple Access (NOMA) to minimize the total system cost [21], and convex programming-driven power allocation via iterative offloading ratio derivation [22]. Game theory applications further enhance system equilibrium through Nash equilibrium solutions [23], a Minority Game (MG)-based scheme and found the optimal solution by probabilistically adjusting decisions, achieving near-optimal performance and stability [24] and a decentralized offloading game algorithm to solve task offloading problems in edge computing [25].

Despite significant advancements in Intelligent Transportation Systems and Connected and Automated Vehicles, current research has not fully addressed the challenges introduced by highly dynamic vehicular environments, diverse service requirements, and heterogeneous resource distributions. A thorough analysis of these issues reveals the necessity for a robust system model and an optimization framework that can effectively balance the conflicting objectives of delay minimization and energy efficiency under stringent constraints. Therefore, designing an effective strategy for task offloading and resource allocation that also minimizes costs presents a significant challenge [26].

In this paper, we propose a three-stage hybrid multi-objective task offloading strategy for UAV-assisted MEC systems to minimize the total system delay and overall energy consumption while ensuring performance under resource constraints. The control decisions include Offload task allocation, UAV selection, and UAV computation resource allocation. The primary contributions of this study are listed below:

- 1. This paper comprehensively considers the computing offloading problem in sustainable UAVs-MEC systems and formulates it as a multi-objective optimization problem, consisting of two optimization objectives, i.e., delay and energy consumption with several constraints.
- 2. This paper proposes a three-stage hybrid multi-objective optimization scheme, called DVCG-MWOA, to address the above-formulated problem. A mobile-aware dynamic clustering algorithm (MDCA) is proposed to dynamically link vehicles and UAVs in the first stage. Then, a cooperative game model with delay constraints is established for each cluster, allocating UAV resources for coordinated utilization. The Pareto front solution is derived using an improved multi-objective whale optimization algorithm.
- 3. We conducted multiple experiments across application scenarios of different scales to thoroughly assess the effectiveness and adaptability of the proposed scheme. Additional parameter analysis reveals that the scheme can flexibly regulate energy consumption, computation delay, and system performance in

response to varying resource constraints and task demands, ultimately enhancing the overall efficiency of task offloading.

## 2 System Model and Problem Formulation

In this section, we detail the UAV-assisted edge computing offloading scheme model, aiming to minimize delay and energy consumption, and formulate the computation offloading problem as a multi-objective optimization problem. Table 1 summarizes the main notations in this model.

Notation Definition	
$\overline{V = \{v_1, v_2, \dots, v_n\}}$	A set of vehicles
$U = \{u_1, u_2, \ldots, u_c\}$	A set of UAVs
$W = \{w_1, w_2, \dots, w_n\}$	A set of tasks
$D_i, C_i, \tau_i$	The task data size, complexity, and tolerable delay of $w_i$
$a_i^{local}, a_i^{edge}, a_i^{uav}$	The proportions of local computing, edge computing and UAV
t t t	computing
$t_{i\_local}, t_{i\_edge}, t_{i\_uav}$	Local computing, edge base station computing, and UAV
0 _	computing delay
$p_{i edge}^{tra}, p_{i uav}^{tra}$	Task transmission power
$E_{i \ local}, E_{i \ edge}, E_{i\_uav}$	Local computing, edge base station computing, and UAV computing
0 _	energy consumption
fv, fedge, fedge_i, fuav, fuav_i	Vehicle computing resources, edge base station computing resources,
	computing resources allocated to vehicle <i>i</i> by the edge base station,
	maximum computing resources of UAV, computing resources allocated
	to vehicle <i>i</i> by UAV
Н	UAV flight altitude
$k_v, k_{edge}, k_u$	Effective switching capacitance of the chip
$v^{tra}$	Task transmission rate
В	The bandwidth
$\mathcal{P}_{i}^{tra}$	The transmission power of vehicle <i>i</i>
$h_i$	Channel gain between task vehicle <i>i</i> and edge base station or UAV
$N_0$	Channel noise power
$WS_i$	The proposed task offloading metric of vehicle <i>i</i>
$w_{mv}, w_{wl}$	The weight of vehicle mobility metric and vehicle offloading efficiency
	requirement
$Q_i$	The communication quality of vehicle <i>i</i>
$MV_i$	Mobility measurement of vehicle <i>i</i>
9	The vehicles within the cluster
$WL_i$	The task volume to be offloaded by vehicle <i>i</i>
$AD_i$	The average distance between vehicle <i>i</i> and other vehicles within the
	cluster.
$LS_i$	The stability of the communication link
$Vns_i$	The number of neighboring vehicles of vehicle <i>i</i>
Arv <sub>i</sub>	The average speed of each vehicle in the cluster

Table 1: Summary of key parameters in the system model

(Continued)

Table 1 (continued)			
Notation	Definition		
$\left[ V_{min}, V_{max} \right]$	The vehicle's speed range		
$\max D_i$	The maximum distance between vehicle <i>i</i> and its neighboring vehicles.		
R, Q, AM	The UAV positions matrix, the vehicle positions matrix and link matrix		
$\mathcal{N},\mathcal{SA},\mathcal{P}\left(\mathcal{S} ight)$	The set of vehicles of each cluster, the sub-resource allocation		
	strategies, and the multi-objective utility value of strategy ${\mathcal S}$		
T	Iterations		
D	The dimension of the target solution		
σ	The smallest prime number		
$Lb_i, Ub_i$	The lower bound and upper bound of the $j_{th}$ dimensional decision		
	variable		
M	The number of whale individuals		
$P_M$	The initial position of the whale population		
$p_{th}$	Dynamic feedback threshold of proposed MWOA		

#### 2.1 System Model

The UAV-assisted vehicular *ad-hoc* network system is shown in Fig. 1. The tasks generated by the vehicles for offloading can be computed by the vehicles themselves, roadside base stations, and deployed UAVs. Vehicles in the area are represented by  $V = \{v_1, v_2, ..., v_n\}$ , UAVs is  $U = \{u_1, u_2, ..., u_c\}$ , and an edge server is deployed. Assuming the vehicles have the same configuration, each vehicle only communicates with one UAV, and the UAVs are at the same height *h* above the vehicles, described using cartesian coordinates. The set of tasks generated by vehicles is  $W = \{w_1, w_2, ..., w_n\}$ , the *i*-th task  $w_i$  is represented by a triplet  $\{D_i, C_i, \tau_i\}$ .

Assuming the tasks are divisible,  $a_i^{local}$ ,  $a_i^{edge}$  and  $a_i^{uav}$ , respectively represent the proportions of tasks computed locally, offloaded to the roadside base station, and offloaded to the UAV, with their sum equal to 1.

The transmission rate between vehicles and the edge base station or UAV can be calculated using Shannon's theorem [27]:

$$v^{tra} = B \log_2 \left( 1 + \frac{p_i^{tra} h_i}{B N_0} \right) \tag{1}$$

## 2.2 Local Vehicle Computing Model

In the local computing process, the amount of data allocated to the vehicle itself can be represented by  $a_i^{local} \times D_i$ , the power consumed in the computing process can be represented by  $k_v \times (f_v)^3$ , the local computing delay and energy consumption are represented by:

$$t_{i\_local} = \frac{C_i \times a_i^{local} \times D_i}{f_v},\tag{2}$$

$$E_{i\_local} = k_{\nu} \times C_i \times a_i^{local} \times D_i \times (f_{\nu})^2.$$
(3)

1)



Figure 1: Model of UAV-assisted Vehicular Ad-Hoc Network (VANET) system

## 2.3 Roadside Base Station Computing Model

In the roadside base station computing process, the amount of data allocated to the roadside base station can be represented by  $a_i^{edge} \times D_i$ , the delay and energy consumption generated by transmitting tasks from the vehicle to the roadside base station represented by:

$$t_{i\_edge}^{tra} = \frac{C_i \times a_i^{edge} \times D_i}{v_{edge}^{tra}},\tag{4}$$

$$E_{i\_edge}^{tra} = p_{i\_edge}^{tra} \times t_{i\_edge}^{tra}.$$
(5)

The computing power used by the roadside base station to process the task is represented by  $k_{edge} \times (f_{edge_i})^3$ . The computing power used by the roadside base station to process the task is represented by:

$$t_{i\_edge}^{mid} = \frac{C_i \times a_i^{edge} \times D_i}{f_{edge\_i}},\tag{6}$$

$$E_{i\_edge}^{mid} = k_{edge} \times C_i \times a_i^{edge} \times D_i \times \left(f_{edge\_i}\right)^2.$$
<sup>(7)</sup>

After the task is processed, the result is returned to the vehicle *i*.

In the roadside base station computing process, the total task processing delay is represented by the sum of the transmission delay and the computing delay, and the total energy consumption is represented similarly.

## 2.4 UAV Computing Model

In the UAV computing process, the amount of data allocated to the UAV can be represented by  $a_i^{uav} \times D_i$  the delay and energy consumption generated by transmitting tasks from the vehicle to the UAV are represented by:

$$t_{i\_uav}^{tra} = \frac{C_i \times a_i^{uav} \times D_i}{v^{tra}},\tag{8}$$

$$E_{i\_uav}^{tra} = p_{i\_uav}^{tra} \times t_{i\_uav}^{up}.$$
(9)

The computing power used by the UAV to process the task is represented by  $k_u \times (f_{uav_i})^3$ , the computing delay, and energy consumption is represented by:

$$t_{i\_uav} = \frac{C_i \times a_i^{uav} \times D_i}{f_{uav\_i}},\tag{10}$$

$$E_{i\_uav} = k_u \times C_i \times a_i^{uav} \times D_i \times (f_{uav})^2.$$
<sup>(11)</sup>

After the task is processed, the result is returned to the vehicle *i*.

In the UAV computing process, the total task processing delay is represented by the sum of the transmission delay and the computing delay, and the total energy consumption is represented similarly.

#### 2.5 Formulation of Problem

The total delay and total energy consumption of the task processing process for vehicle *i* are represented as follows:

$$T_{i\_sum} = t_{i\_local} + t_{i\_edge} + t_{i\_uav}, \tag{12}$$

$$E_{i\_sum} = E_{i\_local} + E_{i\_edge} + E_{i\_uav}.$$
(13)

The problem of local offloading, roadside base station, and UAV edge computing for multiple mobile vehicles in the area is modeled as a multi-objective optimization constraint problem *Z*, and the mathematical model is constructed as follows:

$$Z = \min\left\{\sum_{i=1}^{N} T_{i\_sum}, \sum_{i=1}^{N} E_{i\_sum}\right\},$$
(14)

s.t.

$$a_i^{local} + a_i^{edge} + a_i^{uav} = 1, \forall i \in [1, 2, 3, ..., n],$$
(14a)

$$t_{i\_sum} \le \tau_i, \forall i \in [1, 2, 3, \dots, n],$$
(14b)

$$\sum_{i=1}^{n} f_{edge_i} \leq f_{edge},\tag{14c}$$

where Eq. (14a) indicates the sum of the ratios at which tasks are allocated to different offloading terminals. Eq. (14b) indicates the total processing delay of the task should be lower than the maximum tolerable delay. Eq. (14c) indicates the sum of the computing resources allocated by the edge server to each vehicle should be less than its own computing resources.

#### **3** Proposed Optimization Solution

In this section, which first performs dynamic clustering of vehicles, determines the optimal resource allocation scheme for each cluster head-unmanned aerial vehicle (CH-UAV) through cooperative game theory. To further improve performance, this study develops an enhanced multi-objective whale optimization algorithm to efficiently explore the global optimal solution of the objective function.

#### 3.1 MDCA Clustering Algorithm

The high-speed mobility of vehicular nodes introduces critical IoV challenges including dynamic topology fluctuations, scalability constraints, and routing efficiency demands. Clustering approaches that leverage mobility metrics tend to outperform traditional methods in enhancing cluster stability. Metrics such as vehicle density, relative velocity, relative distance, and moving direction are particularly well-suited for the dynamic environment of VANET system.

In this part, a new clustering algorithm named MDCA that is based on mobility values and vehicle dynamic characteristics is proposed. Considering the mobility and communication quality of each vehicle, dynamically allocate vehicles into clusters with UAVs as cluster heads. Utilize vehicle clustering to enhance UAV offloading efficiency. Continuously adjust UAV positions based on vehicle dynamics at each moment to ensure data processing quality, better addressing the dynamic mobility environment of VANET system.

The following discusses the details of various components and their communication. Vehicles and UAVs serve as the mobile nodes and key elements of our network architecture, and they are equipped with GPS devices. On-board units (OBUs) provide mutual wireless communication via vehicle-to-vehicle (V2V) and vehicle-to-UAV (V2U). A vehicle is assigned to a single cluster at any time slot. Each cluster, headed by a UAV, is responsible for the overall management of the cluster and overseeing the efficient allocation of resources.

We propose a task offloading metric that integrates mobility measurement, task offloading volume, and communication quality as follows:

$$WS_i = w_{mv} \frac{MV_i}{\sum_{q \in Cl} MV_q} + w_{wl} \frac{WL_i}{\sum_{q \in Cl} WL_q} \times \frac{Q_i}{\sum_{q \in Cl} Q_q},$$
(15)

 $Q_i$  is the communication quality between vehicle *i* and the UAV, represented by the reciprocal of the Euclidean distance:

$$Q_i = \frac{1}{Dis(i, UAV)^2}.$$
(16)

Mobility measurement [28]  $MV_i$  is defined as follows:

$$MV_i = \frac{LS_i(t)}{Vns_i} + \sqrt{\left(\ln\left(1 - \frac{Arv_i}{V_{max}}\right)\right)^2 + \frac{AD_i}{maxD_i}}.$$
(17)

The algorithm process is detailed in Algorithm 1. The main objective is to divide *n* vehicles into *c* distinct CH-UAV clusters during each time slot. The location and speed of vehicles can be acquired either through integrated GPS modules or by employing appropriate localization techniques.

1: <b>Input:</b> Input the position matrices R and O of vehicles and UAVs initialize the link matrix AM to 0		
1. Input the position mutilees it and Q of venicles and only s, initialize the link mutik right to o		
<b>Output:</b> the link matrix AM between UAVs and vehicles		
Initialization: Divide the vehicles into the nearest UAV clusters by Euclidean distance and update AM		
For each cluster do		
5: For each vehicle do		
6: Calculate $WS_i$ of each vehicle in the cluster use Eq. (15)		
7: <b>End</b>		
8: Calculate the average value of the dynamic task offloading metrics of the remaining vehicles in the cluster $\overline{WS}$		
9: Update the UAV position to directly above the vehicle with the smallest absolute value $\overline{WS}$ in the		
round of calculations		
10: For each vehicle do		
11: If $v_i$ is in the coverage zone of CH UAV then		
12: If closer in Euclidean distance to other UAV then		
13: update connectivity relationships and matrix <i>AM</i> .		
14: <b>Else</b>		
5: Establishing multi-hop communication with UAV via V2V communication.		
6: End if		
17: Else		
18: If $v_i$ is in the coverage zone of other UAV then		
19: update connectivity relationships and matrix <i>AM</i> .		
20: Else		
21: Establishing multi-hop communication with UAV via V2V communication.		
22: End if		
23: End if		
24: End for		
25: End for		

Initialize the CH-UAV positions and store them in matrix R, and the vehicle positions are stored in matrix Q. Initialize a link matrix AM of size  $x \times n$ ,  $AM_{cd}$  is the link relationship between UAV c and vehicle d. Vehicle d is clustered to UAV c,  $AM_{cd}$  is 1, otherwise,  $AM_{cd}$  is 0. Initially, link each vehicle to the nearest UAV based on Euclidean distance.

The process of dynamic task offloading and communication binding begins by calculating the dynamic task offloading metric WS for each vehicle in a cluster using Eq. (15) and determining the cluster's average  $\overline{WS}$ . The UAV position is then updated to be directly above the vehicle with the smallest absolute WS value. Next, the communication range is assessed: if all vehicles are within the UAV's range, those farther away may select a UAV with a closer Euclidean distance, and the link relationship matrix AM is updated accordingly. If some vehicles are outside the range, their positions are checked for coverage by other UAVs. If covered, links are established with those UAVs, and AM is updated; otherwise, V2V communication is established with the nearest vehicle to form a multi-hop routing connection to the UAV, and the matrix is modified. This process completes the communication binding for the current time slot. Finally, UAV positions are continuously adjusted in real-time based on road conditions to enhance data transmission quality and ensure network stability, with the updated link relationship matrix AM outputted at the end.

#### 3.2 Intra-Cluster Cooperative Game

The algorithm clusters vehicles with offloading tasks based on movement similarity, using UAVs as cluster heads. A coalition formation game coordinates resource allocation within each cluster. To handle unpredictable task demands, the approach employs conditional value-at-risk (CVaR) for risk assessment and introduces a negotiation mechanism to optimize allocation while ensuring task completion within tolerable latency. By integrating consensus auctions, the approach maintains UAV autonomy and stability in the coalition structure. The UAV resource allocation problem as a cooperative game model among the vehicles:

$$\mathcal{G} = \{\mathcal{N}, \mathcal{SA}, \mathcal{P}(\mathcal{S})\},\tag{18}$$

where  $\mathcal{N}$  is the set of all vehicles of each cluster,  $S\mathcal{A}$  is the set of all sub-resource allocation strategies, and  $\mathcal{P}(S)$  represents the multi-objective utility function of strategy S, which is the sum of the delay and energy consumption generated by q vehicles in the cluster.

$$\mathcal{P}(\mathcal{S}) = \{\sum_{i=1}^{q} T_i, \sum_{i=1}^{q} E_i\}.$$
(19)

**Definition 1 (Coalition structure):** The coalition structure  $\prod = \{S_0, S_1, ..., S_q\}$ , is a disjoint partition of the vehicleresource allocation set. Each coalition  $\prod \in SA$  is specific to the subresourceallocationset. Each element  $S_k$  represents the proportion of UAV resources allocated to vehicle k.

**Definition 2 (Negotiation mechanism):** For vehicle  $k \in N$ , if vehicle k changes its strategy, it needs to satisfy the conditions:

$$T_k(f_{uav-k}') \le T_k^{\tau} \cup E_i(f_{uav-k}') \le E_k(f_{uav-k}), \forall k \in \mathcal{N},$$
(20)

 $\mathcal{P}(\mathcal{S}') < \mathcal{P}(\mathcal{S}),\tag{21}$ 

where S' represents the changed strategy.  $f_{uav-k}$ ' represents the UAV computing capacity allocated to vehicle k after the strategy change.  $T_k^{\tau}$  represents the maximum tolerable delay for the UAV processing the task portion, expressed as the product of the proportion of the task processed by the UAV and the maximum tolerable delay of the task, i.e.,  $a_k^{uav} \times \tau_k$ .

The UAV resource allocation game algorithm process is shown in Algorithm 2. In the initial stage, input the link matrix *AM* of vehicles and UAVs from the clustering stage, and the task information to be allocated for each vehicle.

Alg	orithm 2: UAV resource allocation game algorithm	
1:	Input: UAV and vehicle link matrix task information for each vehicle	
2:	<b>Output:</b> Optimal resource allocation strategy	
3:	Initialization: Evenly distribute the UAV's computing resources among the vehicles in the cluster	
4:	For each cluster do	
5:	For each vehicle do	
6:	If the minimum latency requirement is satisfied then	
7:	Gradient descent reduces resource allocation use (23)	
8:	Else if	
9:	Gradient ascent increases resources allocated use (23)	
	(Continued)	

Algorithm 2 (continued)		
10:	End	
11:	If (20) & (21) true then	
12:	Update the optimal resource allocation strategy	
13:	End	

In the initial iteration, the UAV's computing power is evenly distributed among the vehicles in the cluster. Determine whether each task meets the delay tolerance requirement. For tasks that meet the delay requirement, use the gradient descent method to reduce computing power allocation; for tasks that do not meet the delay requirement, use the gradient ascent method to increase computing power allocation. Continuously iterate under the constraints of the negotiation mechanism to seek the optimal strategy set  $S^*$ .

In this cooperative game model, delay is used as a constraint for the optimization goal. Gradient calculations are performed based on energy consumption to derive the optimal utility strategy. The complexity of each vehicle's task, data size, task period, and switching capacitance values are set as constants  $o_k$ ,  $E_k$  denoted as:

$$E_k = o_k \times (\mathcal{S}_k \times f_{uav})^2, \tag{22}$$

where  $o_k$  is  $k_u \times C_k \times a_k^{uav} \times D_k$ .

Computing the partial derivatives  $S_k$  to calculate the gradient.

Update the strategy of each vehicle in the direction opposite to the gradient using the stochastic gradient algorithm, ensuring it continuously meets the conditions of (20) and (21). The update rule is as follows:

$$\mathcal{S}_{k}^{(I+1)} = \mathcal{S}_{k}^{(I)} \pm \eta \times \nabla E(\mathcal{S}_{k}).$$
<sup>(23)</sup>

Define  $S_k^{(I+1)}$  as the strategy for the next iteration, continue iterating until the optimal strategy for the cooperative game model is obtained. Output the proportion of UAV resources allocated.

## 3.3 Proposed MWOA Offloading Optimization Strategy

In multi-objective optimization (MOO), the inherent conflict between objectives creates a Pareto optimal set where improving one metric necessitates compromising others. While the Whale Optimization Algorithm (WOA) has emerged as an effective MOO solver due to its exploration-exploitation balance and computational efficiency, existing multi-objective WOA variants still face limitations in handling complex vehicular scenarios. To address this, we propose an enhanced MOWOA integrating swarm-based whale search with evolutionary mechanisms. Each whale solution encodes vehicle/UAV capabilities and task offloading ratios across three layers (vehicles, RSUs, UAVs), with dual optimization objectives of latency and energy consumption. As detailed in Algorithm 3, two critical improvements are implemented over baseline MOWOA to achieve parallel optimization.

Algorithm 3: Offloading strategy optimization using improved MOWOA

- 1: **Input:** *AM*, task information for each vehicle, UAV resource allocation strategy, Population size *M*, Iterations *T*
- 2: **Output:** Pareto optimal set
- 3: **Initialize** good point set for the population
- 4: For each dimension j in D do

## Algorithm 3 (continued)

5:	Select the smallest prime number $\sigma$		
6:	Generate good point set <i>P</i> using (25)		
7:	End for		
8:	Map the good point set $P$ to the search space using (26)		
9:	Initialize external archive to store non-dominated solutions		
10:	while $t < T$ do		
11:	Calculate fitness values for each whale individual		
12:	Update external archive with non-dominated solutions		
13:	Perform non-dominated sorting on the population		
14:	Calculate crowding distance for each whale individual		
15:	Sort whale individuals by Pareto level and crowding distance		
16:	Select leaders from external archive		
17:	For each whale individual do		
18:	Update position using dynamic feedback threshold		
19:	If first iteration then		
20:	Set $p_{th}$ is 0.5		
21:	Else		
22:	Update threshold using Eq. (27)		
23:	End if		
24:	Update position based on spiral and contraction mechanisms with updated threshold		
25:	End for		
26:	Update $t = t + 1$		
27:	End while		
28:	Return Pareto optimal set from external archive		

The process is as follows:

Step 1: Good point set initialization population:  $G_D$  is a unit cube in the *D* dimensional Euclidean space, *D* represents the dimension of the target solution, points  $\gamma$  in the Euclidean space can be represented by *D* dimensional vectors, i.e.,  $\gamma = (\gamma_1, \gamma_2, ..., \gamma_D)$ ,  $\gamma \in G_D$ , where  $0 \le \gamma_j \le 1$ , be represented as:

$$\gamma_j = 2\cos\left(\frac{2j\pi}{\sigma}\right), 1 \le j \le D,\tag{24}$$

where *j* is a dimension in *D*,  $\sigma$  is the smallest prime number that satisfies  $\frac{(\sigma-3)}{2} \ge D$ , generating the good point set  $P_M(m)$ :

$$P_{\mathrm{M}}(\mathrm{m}) = \{(\gamma_{1} \times \mathrm{m}), \dots, (\gamma_{D} \times \mathrm{m})\}, 1 \le \mathrm{m} \le \mathrm{M},$$

$$(25)$$

where *M* is the number of whale individuals in the population, m = 1, 2, ..., M, with m representing an individual whale in the population,  $\gamma_j \times m$  represents a good point,  $P_M(m)$  is the good point set matrix of size  $D \times M$ , which is mapped to the search space as:

$$P_{\mathrm{M}}(\mathrm{m}) = Lb_{j} + (Ub_{j} - Lb_{j}) \times \mathrm{mod}\left(\left\{P_{\mathrm{M}}(\mathrm{m})\right\}, 1\right), 1 \le j \le D,$$
(26)

where  $Lb_j$  is the lower bound of the  $j_{th}$  dimensional decision variable  $Ub_j$  is the upper bound of the  $j_{th}$  dimensional decision variable.  $P_M(m)$  mapped is the initial position of the whale population.

Step 2: Establish Pareto front storage mechanism: An external archive stores Pareto non-dominated solutions, replacing lower-fitness ones when capacity is exceeded to ensure dynamic updates.

Step 3: Non-dominated sorting: Individuals are ranked based on dominance, selecting non-dominated ones as the first level and iterating until all are classified.

Step 4: Crowding distance sorting mechanism: to maintain diversity and explore the solution space, crowding distance guides the algorithm toward the Pareto front by measuring distribution density. Solutions with greater distances are prioritized to ensure uniform coverage and enhance optimization.

Step 5: Elite retention mechanism: whale individuals from parent and offspring generations are sorted by crowding distance within each Pareto level. Those with the largest distances are stored in the archive until full, while the rest are discarded.

Step 6: Leader selection strategy: a whale individual with the smallest Pareto rank is randomly selected from the archive to guide the group's movement in each iteration.

Step 7: Enhancement of individual location update strategy: the fixed 0.5 selection probability in the original whale optimization algorithm causes an imbalance between global exploration and local exploitation, slowing convergence. We propose a dynamic feedback threshold mechanism to adjust these probabilities, improving both convergence speed and optimization performance. The selection probability  $p_{th}$  is updated dynamically as follows:

$$p_{th}(t+1) = \begin{cases} p_{th}(t) + (ra/M), & ra > 0\\ 1 - p_{th}(t), & ra = 0 \end{cases},$$
(27)

where *M* is the total number of whale individuals in the population, and *ra* represents the number of populations whose fitness values have improved compared to the previous iteration. At the first iteration,  $p_{th}$  is set to 0.5, and in subsequent iterations,  $p_{th}(t+1)$  is updated based on the value of  $p_{th}$  from the previous iteration.

The rest of the process is updated according to the WOA.

#### 3.4 Method Overview

This paper considers the task offloading and resource allocation problem as a multi-objective optimization problem, aiming to minimize latency and energy consumption. Firstly, dynamic clustering is formed through the current positions and mobility characteristics of vehicles and UAVs, enabling cooperation in each time slot. Subsequently, a cooperative game model constrained by latency is used for intra-cluster resource allocation. To find the optimal task assignment strategy and resource utilization, an improved parallel MOWOA algorithm is employed for multi-objective optimization. This scheme is potentially very effective in balancing the trade-off between latency and energy consumption, providing efficient solutions in complex environments. The proposed scheme process is shown in Fig. 2.

#### **4** Experimental Evaluation

In this section, a simulation is performed to evaluate the effectiveness of the proposed joint strategy. The experiment is conducted in MATLAB R2022b, and run on a personal computer with Intel (R) Core (TM) i9-13900 CPU and 32.0 GB RAM. The results are based on the average of 30 Monte Carlo simulations.



Figure 2: The flow chart of DVCG-MWOA

#### 4.1 Experimental Environment

In our experiment, we simulate an intersection composed of two perpendicular roads, each 500 m in length. A roadside base station is set on one side of the road, and drones are randomly distributed in the area. Five experimental vehicle scales are set up, with vehicles randomly generated according to a Poisson distribution. The clustering cycle between drones and vehicles is 3 s (ignoring the time required for drone movement), and task offloading is performed every 6 s. The experiment runs for 6 min, and the results are averaged over 10 task offloads. The experimental parameters set in this paper are shown in Table 2. The communication standard employed in our experiments is the IEEE 802.11p standard. Specifically, simulation map is based on [29], the vehicle speed configuration references [30], the task complexity and chip switching capacitance as suggested in [31].

Parameter	Default value
Data Size D <sub>i</sub>	[50, 200] KB
Task Complexity $C_i$	[100, 1000] cycles/bit
Maximum Tolerable Delay $\tau_i$	[250, 1500] ms
Vehicle Computing Capability $f_v$	$[5 \times 10^8, 7 \times 10^8]$ cycles/bit
UAV Computing Capability $f_{uav}$	$[6 \times 10^8, 8 \times 10^8]$ cycles/bit
The Vehicle's Speed Range $[V_{min}, V_{max}]$	[60, 80] km/h
UAV Flight Height H	60 m
The number of Vehicles <i>n</i>	10, 20, 30, 40, 50

(Continued)

## Table 2 (continued)

Parameter	Default value
The number of UAVs <i>c</i>	5, 10, 15
Edge Server Computing Capability $f_{edge}$	$5 \times 10^{10}$ cycles/bit
Chip Switching Capacitance $k_v$ , $k_{edge}$ , $k_u$	$1 \times 10^{-26}$ , $1 \times 10^{-26}$ , $1 \times 10^{-27}$
Transmission Power When Vehicles Send Tasks	5 W
$p_i^{tra}$	
Channel Gain Between Task Vehicles and Edge	$1 \times 10^{-4} \text{ Hz}$
Base Stations or UAVs $h_i$	
Communication Bandwidth B	$5 \times 10^7 \text{ Hz}$
Channel Noise Power $N_0$	$5 \times 10^{-14} \mathrm{W}$

## 4.2 Task Offloading Strategies

### 4.2.1 Benchmark Offloading Location Schemes

In the simulation experiments, we mainly study the vehicles, edge servers, and UAVs to collaborate for computation task offloading called Local, Edge and Air Computation Offloading (LEACO), This scheme was compared with the following three benchmark offloading location schemes.

- 1. LCO (Local Computation Only) scheme, where tasks are only offloaded locally.
- 2. ECO (Edge Computation Only) scheme, where tasks are only offloaded to edge servers.
- 3. LECO (Local and Edge Computation Offloading) scheme, where tasks are offloaded both locally and to edge servers.

#### 4.2.2 Performance Metrics

In our experiments, to facilitate comparison with single-objective experiments, we not only considered delay and energy consumption separately but also considered the weighted sum of these two normalized optimization objectives to represent the total system cost. The definitions are as follows.

$$Q = \alpha \times Delay_w + (1 - \alpha) \times Energy_w,$$
(28)

where  $Delay_w$  and  $Energy_w$  represent the normalized delay and normalized energy consumption, respectively, and  $\alpha$  represents the weight factor. Particularly, two normalized objective values  $Delay_w$  and  $Energy_w$  are calculated as follows.

$$Delay_w = \frac{2 \times Delay_{all}}{Delay_{all}^L + Delay_{all}^E},$$
(29)

$$Energy_w = \frac{2 \times Energy_{all}}{Energy_{all}^L + Energy_{all}^E},$$
(30)

where  $Delay_{all}^{L}$  and  $Delay_{all}^{E}$  represent the total delay when all tasks are executed in LCO scheme and all tasks are executed in ECO scheme, respectively.  $Energy_{all}^{L}$  and  $Energy_{all}^{E}$  represent the total energy consumption when all tasks are executed in LCO scheme and all tasks are executed in ECO scheme, respectively. These two objective values can be in the same order of magnitude, so the weighted sum can simultaneously reflect the algorithm's comprehensive performance in terms of delay and energy consumption.

## 4.3 Comparative Optimization Approaches

We compare the proposed joint scheme with the following methods, which combine offloading methods and optimization algorithms.

#### 4.3.1 Offloading Methods

- 1. Random Sequential Scheme (RS): Vehicles are randomly bound to a UAV, which then offloads tasks sequentially.
- 2. Distance-First Averaging Scheme (DA): Vehicles bind only with the nearest UAV, and the UAV's computational resources are evenly distributed.
- 3. Cyclic Balanced Scheme (CB): Sort all vehicle offloading tasks by size, then sequentially assign these sorted tasks to the available UAVs with the most idle computing resources for computation.

#### 4.3.2 Optimization Algorithms

We also compare these offloading strategies using several intelligent optimization algorithms to evaluate their effectiveness in minimizing system costs:

- 1. Single-objective Comparison: Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Improved Whale Optimization Algorithm (IWOA).
- 2. Multi-objective Comparison: Non-dominated Sorting Genetic Algorithm III (NSGA-III), Multi-objective Pareto Envelope-based Selection Algorithm II (PESA-II) and Multi-objective Whale Optimization Algorithm (MOWOA).

By integrating these intelligent optimization algorithms with the offloading methods, we aim to compare their performance with our proposed joint scheme, focusing on finding the optimal system cost.

#### 4.4 Simulation Results

Fig. 3 shows we conducted a set of experiments to explore the optimal alpha value and number of UAVs in the LEACO scenario, laying the foundation for subsequent research. In terms of methodology, we adopted a difference-based optimization strategy, selecting the alpha value and UAV quantity that minimized the total system cost difference, thereby effectively reducing subjective bias in the experimental process.



**Figure 3:** The variation in the absolute difference in costs between the LECO and LEACO offloading modes under different numbers of UAVs, as the value of alpha changes for different vehicle quantities

This approach not only validated the effectiveness of the LEACO scenario but also ensured the reliability and scientific rigor of the experimental results. Specifically, we conducted experiments in both LEACO and LECO scenarios under different alpha values (0.45, 0.55, 0.65, 0.75) and varying numbers of UAVs (5, 10, 15). For each configuration, we calculated the difference in total system cost for different vehicle quantities (10, 20, 30, 40, 50). As shown in the figure, when the alpha value is 0.75 and the number of UAVs is 10, the system cost difference reaches its minimum, and thus we follow this indicator for subsequent experiments. Optimizing the system cost difference under complex and variable conditions ensures that the chosen alpha value and number of UAVs demonstrate significant feasibility and robustness in practical applications. Consequently, this provides a solid theoretical foundation and a reliable experimental basis for subsequent studies.

Fig. 4 shows the experimental results of DVCG-MWOA compared to single-objective hybrid comparison schemes in terms of delay and energy consumption weight, we calculated the average value of the component in the Pareto solution set. As shown in the figure, across five different experimental environments with varying vehicle quantities, the proposed hybrid strategy reduced the time delay weights by an average of 27.63%, 18.02%, and 8.71% compared to the single-objective hybrid strategies (RS-PSO, DA-GA, CB-IWOA), and reduced the energy weights by an average of 22.12%, 33.69%, and 10.51% compared to the single-objective hybrid strategies, respectively. DVCG-MWOA shows clear advantages compared to these single-objective comparison algorithms.



**Figure 4:** The delay weight comparison results and energy weight comparison results with the single target experiment under different experimental sizes

Fig. 5 compares the Pareto frontiers of the proposed DVCG-MWOA with six baseline combinations of offloading methods and optimization algorithms under varying vehicular densities (10/30/50 vehicles). The hybrid baselines exhibit distinct limitations: RS-NSGA-III induces latency spikes and energy fluctuations due to uncoordinated task allocation and stochastic vehicle-UAV binding, while DA-PESA-II suffers from progressive performance decay as its static proximity-based binding exacerbates congestion at nearest UAVs. Although CB-MOWOA mitigates load imbalance through sorted task scheduling, its rigid cyclic allocation incurs computational overheads under high-density scenarios (50 vehicles), leading to suboptimal resource utilization. In contrast, DVCG-MWOA integrates mobility-aware game-theoretic coordination with latency-constrained optimization, dynamically adjusting UAV-vehicle binding based on real-time trajectories. This dual adaptation mechanism reduces total system costs by 22%–35% in Pareto dominance

metrics, while maintaining stable energy-delay trade-offs even under extreme densities. Furthermore, it outperforms single/multi-objective baselines in convergence precision (12.7%–29.4% improvement) and solution diversity, demonstrating superior adaptability to dynamic vehicular environments.



Figure 5: The comparison results with multi-objective algorithm experiments under different numbers of vehicles

Fig. 6 illustrates the task offloading success rates of different strategies under varying maximum tolerable delays, with vehicle counts set to 20 and 40, respectively. The results indicate a positive correlation between the probability of successful task offloading and the tolerable delay. The RS-NSGA-III shows a gradual increase in offloading rates as the delay increases, but overall, its performance remains conservative, particularly with a higher number of vehicles. In comparison, the DA-PESA-II achieves slightly higher offloading rates across all delay conditions, although its growth trend is relatively moderate. The CB-MOWOA exhibits a stable growth trend, performing particularly well under high delay conditions; it achieves a high offloading rate at 750 ms, with smaller increases thereafter. Among the schemes, the DVCG-MWOA consistently outperforms the others under all delay conditions, nearing saturation in high delay scenarios and showing superior performance across both vehicle scales.



**Figure 6:** The comparison results in maximum tolerable delay and task offloading success rate when the number of vehicles is 20 and 40, respectively

## **5** Conclusion

In this paper, we comprehensively study sustainable edge computing in the Internet of Vehicles, considering the complex and variable vehicle conditions, reasonable resource allocation optimization, and UAV-assisted edge computing issues in practical scenarios of the Internet of Vehicles. We formulate the computation offloading problem as a multi-objective optimization problem with the goal of minimizing latency and energy consumption. Further, we design a three-stage hybrid offloading strategy to effectively solve the above problems. The first step is to cluster the vehicles, followed by resource allocation within the clusters, and then solve using an improved parallel multi-objective whale optimization algorithm. In this way, we achieve the goal of reducing latency and energy consumption.

Acknowledgement: The authors sincerely appreciate the editor and reviewers for their valuable feedback.

**Funding Statement:** This research was funded by Shandong University of Technology Doctoral Program in Science and Technology, grant number 4041422007.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, methodology and writing, Yiwei Zhang; supervision and project administration, Xin Cui; formal analysis, Qinghui Zhao. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available from the corresponding author, Xin Cui, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

## References

- 1. Matin A, Dia H. Impacts of connected and automated vehicles on road safety and efficiency: a systematic literature review. IEEE Trans Intell Transp Syst. 2023;24(3):2705–36. doi:10.1109/TITS.2022.3227176.
- 2. Shi J, Du J, Shen Y, Wang J, Yuan J, Han Z. DRL-based V2V computation offloading for blockchain-enabled vehicular networks. IEEE Trans Mob Comput. 2023;22(7):3882–97. doi:10.1109/TMC.2022.3153346.
- 3. Wang J, Zhu K, Hossain E. Green Internet of vehicles (IoV) in the 6G era: toward sustainable vehicular communications and networking. IEEE Trans Green Commun Netw. 2022;6(1):391–423. doi:10.1109/TGCN.2021. 3127923.
- 4. Liu L, Chen C, Pei Q, Maharjan S, Zhang Y. Vehicular edge computing and networking: a survey. Mob Netw Appl. 2021;26(3):1145–68. doi:10.1007/s11036-020-01624-1.
- 5. Mozaffari M, Saad W, Bennis M, Nam YH, Debbah M. A tutorial on UAVs for wireless networks: applications, challenges, and open problems. IEEE Commun Surv Tutor. 2019;21(3):2334–60. doi:10.1109/COMST.2019.2902862.
- 6. Dai X, Xiao Z, Jiang H, Lui JCS. UAV-assisted task offloading in vehicular edge computing networks. IEEE Trans Mob Comput. 2024;23(4):2520–34. doi:10.1109/TMC.2023.3259394.
- 7. Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Trans Intell Transp Syst. 2022;23(2):740–59. doi:10.1109/TITS.2020.3024655.
- Mollah MB, Zhao J, Niyato D, Guan YL, Yuen C, Sun S, et al. Blockchain for the Internet of vehicles towards intelligent transportation systems: a survey. IEEE Internet Things J. 2021;8(6):4157–85. doi:10.1109/JIOT.2020. 3028368.
- 9. Noor-A-Rahim M, Liu Z, Lee H, Khyam MO, He J, Pesch D, et al. 6G for vehicle-to-everything (V2X) communications: enabling technologies, challenges, and opportunities. Proc IEEE. 2022;110(6):712–34. doi:10.1109/JPROC. 2022.3173031.
- 10. Wang X, Ning Z, Guo S, Wang L. Imitation learning enabled task scheduling for online vehicular edge computing. IEEE Trans Mob Comput. 2022;21(2):598–611. doi:10.1109/TMC.2020.3012509.

- Mustafa E, Shuja J, Rehman F, Namoun A, Bilal M, Bilal K. Deep reinforcement learning and SQP-driven task offloading decisions in vehicular edge computing networks. Comput Netw. 2025;262:11180. doi:10.1016/j.comnet. 2025.111180.
- 12. Mustafa E, Shuja J, Rehman F, Namoun A, Bilal M, Iqbal A. Computation offloading in vehicular communications using PPO-based deep reinforcement learning. J Supercomput. 2025;81(4):547. doi:10.1007/s11227-025-07009-z.
- 13. Yang X, Luo H, Sun Y, Guizani M. A novel hybrid-ARPPO algorithm for dynamic computation offloading in edge computing. IEEE Internet Things J. 2022;9(23):24065–78. doi:10.1109/JIOT.2022.3188928.
- 14. Maleki H, Başaran M, Durak-Ata L. Handover-enabled dynamic computation offloading for vehicular edge computing networks. IEEE Trans Veh Technol. 2023;72(7):9394–405. doi:10.1109/TVT.2023.3247889.
- 15. Zhao L, Li T, Zhang E, Lin Y, Wan S, Hawbani A, et al. Adaptive swarm intelligent offloading based on digital twin-assisted prediction in VEC. IEEE Trans Mob Comput. 2024;23(8):8158–74. doi:10.1109/TMC.2023.3344645.
- 16. Shen R, Gao M, Li W, Li Y. Dynamic task offloading in distributed VEC networks: an exploration and exploitation assisted contract-theoretic approach. IEEE Trans Veh Technol. 2024;73(4):5717–29. doi:10.1109/TVT.2023.3332956.
- 17. Raza S, Wang S, Ahmed M, Anwar MR, Mirza MA, Khan WU. Task offloading and resource allocation for IoV using 5G NR-V2X communication. IEEE Internet Things J. 2022;9(13):10397–410. doi:10.1109/JIOT.2021.3121796.
- 18. Lv W, Yang P, Zheng T, Yi B, Ding Y, Wang Q, et al. Energy consumption and QoS-aware co-offloading for vehicular edge computing. IEEE Internet Things J. 2023;10(6):5214–25. doi:10.1109/JIOT.2022.3221966.
- 19. Zhan W, Luo C, Wang J, Wang C, Min G, Duan H, et al. Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing. IEEE Internet Things J. 2020;7(6):5449–65. doi:10.1109/JIOT.2020.2978830.
- 20. Shen Q, Hu BJ, Xia E. Dependency-aware task offloading and service caching in vehicular edge computing. IEEE Trans Veh Technol. 2022;71(12):13182–97. doi:10.1109/TVT.2022.3196544.
- 21. Du J, Sun Y, Zhang N, Xiong Z, Sun A, Ding Z. Cost-effective task offloading in NOMA-enabled vehicular mobile edge computing. IEEE Syst J. 2023;17(1):928–39. doi:10.1109/JSYST.2022.3167901.
- 22. Zhao M, Yu JJ, Li WT, Liu D, Yao S, Feng W, et al. Energy-aware task offloading and resource allocation for timesensitive services in mobile edge computing systems. IEEE Trans Veh Technol. 2021;70(10):10925–40. doi:10.1109/ TVT.2021.3108508.
- 23. He Q, Cui G, Zhang X, Chen F, Deng S, Jin H, et al. A game-theoretical approach for user allocation in edge computing environment. IEEE Trans Parallel Distrib Syst. 2020;31(3):515–29. doi:10.1109/TPDS.2019.2938944.
- 24. Hu M, Xie Z, Wu D, Zhou Y, Chen X, Xiao L. Heterogeneous edge offloading with incomplete information: a minority game approach. IEEE Trans Parallel Distrib Syst. 2020;31(9):2139–54. doi:10.1109/TPDS.2020.2988161.
- 25. Zhan Y, Guo S, Li P, Zhang J. A deep reinforcement learning based offloading game in edge computing. IEEE Trans Comput. 2020;69(6):883–93. doi:10.1109/TC.2020.2969148.
- 26. Gong Y, Yao H, Wu D, Yuan W, Dong T, Yu FR. Computation offloading for rechargeable users in space-air-ground networks. IEEE Trans Veh Technol. 2023;72(3):3805–18. doi:10.1109/TVT.2022.3217079.
- 27. Luo Y, Wang Y, Lei Y, Wang C, Zhang D, Ding W. Decentralized user allocation and dynamic service for multi-UAV-enabled MEC system. IEEE Trans Veh Technol. 2024;73(1):1306–21. doi:10.1109/TVT.2023.3308589.
- 28. Salim A, Khedr AM, Osamy W. IoVSSA: efficient mobility-aware clustering algorithm in Internet of vehicles using sparrow search algorithm. IEEE Sens J. 2023;23(4):4239–55. doi:10.1109/JSEN.2022.3233903.
- 29. Zhong R, Liu X, Liu Y, Chen Y. Multi-agent reinforcement learning in NOMA-aided UAV networks for cellular offloading. IEEE Trans Wirel Commun. 2022;21(3):1498–512. doi:10.1109/TWC.2021.3104633.
- 30. Liu L, Zhao M, Yu M, Ahmad Jan M, Lan D, Taherkordi A. Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks. IEEE Trans Intell Transp Syst. 2023;24(2):2169–82. doi:10.1109/TITS.2022.3142566.
- Chen Y, Li K, Wu Y, Huang J, Zhao L. Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: a distributed online approach. IEEE Trans Mob Comput. 2024;23(8):8129–42. doi:10.1109/ TMC.2023.3346431.