

UAV-Assisted Content Caching for Human-Centric Consumer Applications in IoV

Wen Wang, Xiaolong Xu* , *Senior Member, IEEE*, Muhammad Bilal, *Senior Member, IEEE*, Maqbool Khan, and Yizhou Xing

Abstract—With various consumer electronics deployed in Internet of Vehicles (IoV), human-centric consumer in-vehicle applications (e.g., driver assistance, path planning, and healthcare system) can supply high-quality driving experience and enhance travel safety within a short time. In addition, Unmanned Aerial Vehicles (UAV) are expected to be critical to assist terrestrial vehicular networks in delivering delay-sensitive contents of services. However, due to the mutual coupling of trajectory planning of UAVs, serving the same task requests repeatedly in the same area results in wasted resources. Hence, it is challenging to supply high-quality services while ensuring energy-efficient content caching. To solve this dilemma, a content Caching scheme with Trajectory design through differential evolution and Deep Reinforcement learning (CTDR) is introduced. Specifically, a content caching scheme based on differential evolution (DE) is first proposed. Next, a trajectory design optimization based on multi-agent proximal policy optimization (MAPPO) is designed to minimize system energy consumption. Eventually, the superiority of CTDR is demonstrated through various simulated experiments.

Index Terms—IoV, consumer electronics, trajectory design, content caching, UAV, multi-agent reinforcement learning.

I. INTRODUCTION

CONSUMER electronics have played a crucial role in improving the quality of service (QoS) of consumers benefiting from advancements in Artificial Intelligence (AI) and the Internet of Things (IoT). The rapid development of IoT has greatly promoted the continuous growth of consumer electronics, contributing 27% to the increase of the consumer market [1]. Besides, modern consumer electronics have given rise to various human-centric consumer applications to satisfy the requirements of consumers. These applications can demonstrate the validity of modern consumer electronics and fuel the expansion of the consumer market [2].

*Corresponding author

Wen Wang is with the School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: 202183290225@nuist.edu.cn.

Xiaolong Xu is with the School of Software, and Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAET), Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: njxlxu@gmail.com.

Muhammad Bilal is with School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, United Kingdom. E-mail: m.bilal@ieee.org.

Maqbool Khan is with Software Competence Center Hagenberg (SCCH), Austria. E-mail: maqbool.khan@scch.at.

Yizhou Xing is with the Reading Academy, Nanjing University of Information Science and Technology, Nanjing 210044, China. E-mail: 202183060013@nuist.edu.cn.

The Internet of Vehicles (IoV) stands as a prominent domain reaping the advantage of these human-centric applications supported by consumer electronics systems. Specifically, vehicles now are tending to come with operating systems much like a smartphone or laptop computer. They are connected to various consumer electronics devices with transmitting and receiving functions, such as smart cameras, infrared sensors, etc. This interconnected network facilitates enable communication between vehicles and public networks [3]. The integration among these devices forms a novel human-centric IoV ecosystem of interdependence and mutual benefit. This real-time delivery helps improve traffic safety and optimizes the user experience [4]. However, due to the limited computing efficiency of vehicles, they cannot solve most tasks by themselves and need to offload services to cloud servers for processing. This rigid solution usually results in unnecessary transmission energy consumption and increases the time delay [5].

Mobile edge computing (MEC), widely recognized as a potentially capable technology for providing cloud computing services deployed at the network's border, can utilize intelligent computing and data processing technologies to achieve real-time service response to support edge human-centric consumer applications [6]. The node selection strategy in traditional distributed training methods is usually applicable to edge nodes with non-heterogeneous data and resources, which can be implemented without considering the constraints of fixed resources during model training as well as model aggregation [7]. However, when working during peak hours, the edge base station may be overloaded with a massive amount of traffic data. Additionally, MEC deployed on cellular wireless networks as described above is dependent on existing fixed infrastructures on the ground, which are costly to deploy and do not work well in hotspots with time-varying loads or in remote areas without adequate ground infrastructures [8]. Recently, MEC assisted by UAVs has attracted much attention for UAV's advantages of high mobility and rapid deployment. UAVs can help dynamic networks achieve high-speed service transmission and stability in network topology when the fixed edge servers are overloaded or unavailable [9]. Specifically, UAVs can supply computing services to high-speed moving vehicles flexibly for their high mobility, line-of-sight (LoS) link connectivity, and long-range flight capabilities [10], [11].

Although UAVs could greatly alleviate network congestion, it is still a challenge to fully utilize UAVs in the IoV network to provide high-quality services. First, with the constrained storage capacity, it becomes impractical for UAVs to concurrently cache all content files. Moreover, repeatedly downloading sim-

ilar service contents from the remote cloud results in a massive waste of time and computation resources [12]. Second, since every UAV has a finite amount of energy, it is impossible to continuously provide efficient services for all applications within a long period. Third, the number and location of UAVs are numerous and variable, and the communication between them and the ground servers faces multiple impacts such as time delay, frequency shift, and user interference [13]. It remains a significant problem in the caching decision and UAV trajectory design. Therefore, which services need to be cached on the UAV and how to design UAV trajectory in UAV-enabled IoV become the critical problems to be solved.

Furthermore, vehicles move dynamically at a high speed so it is hard to predict their precise locations in a short time. Technically, Markov Decision Process (MDP) has been used extensively to simulate the UAV trajectory, which can be solved by Deep Reinforcement Learning (DRL). It employs deep neural networks to approximate complicated functions and offers a potent framework for solving intricate problems with high-dimensional input spaces. This makes it suitable for scenarios where traditional methods struggle due to their limited representational capacity. However, since traditional single-agent approaches might overlook the collaborative or competitive dynamics that arise when agents influence one another's decisions, they are infeasible for a dynamically changing environment with high-speed UAVs and vehicles [14]. Multi-agent reinforcement learning (MARL), by considering collective interactions and strategic decision-making of multiple agents, can better capture the nuanced dynamics present in complex systems. Thus, MARL has been applied in trajectory design and many studies have achieved remarkable results [15]–[17]. Nevertheless, when the number of UAVs grows or UAV computational resources are limited, the above-mentioned excellent works produce some problems such as over-fitting and slow convergence. Hence, we introduce CTDR for UAV-assisted content caching for human-centric consumer applications in IoV. Specifically, we first utilize differential evolution (DE) to establish the caching scheme. Then, we employ a multi-agent proximal policy optimization (MAPPO) based trajectory design method with a trained caching policy to reduce system energy consumption. Following are the primary contributions of our work:

- Construct a UAV-enhanced content caching framework for human-centric consumer applications in IoV, which is formulated as a mixed integer non-linear programming (MINLP) optimization problem aimed at minimizing energy consumption. Then, the optimization problem is decomposed into two subproblems, namely, content caching scheme and trajectory design strategy.
- Design a human-centric content caching scheme based on DE for consumer applications. UAVs act as cache units to dynamically update cached content with historical data to improve the cache hit ratio.
- Propose a multi-UAV cooperative trajectory optimization strategy, which leverages MAPPO to minimize system energy consumption. Each UAV assumes the role of an autonomous agent to determine the trajectory design

aimed at attaining lower energy consumption.

- Conduct adequate experiments based on simulated environments to evaluate the effectiveness of the CTDR in improving the cache hit ratio and decreasing energy consumption.

Other content of the paper includes: In Section II, we discuss the current research efforts. In Section III, the system model is introduced and the optimization problem is meticulously defined. Section IV details the UAV-assisted content caching scheme CTDR. The efficacy of CTDR is scrutinized in Section V. Finally, Section VI concludes the research, which involves the possible future work.

II. RELATED WORKS

Recently, UAV-assisted MEC has attracted considerable attention, and many works have been conducted to improve communication performance through UAV caching optimization. Specifically, UAVs function as cache units, catering to frequently requested content for their respective users. This approach aims to enable UAVs to promptly deliver service outcomes with minimal delays. Luo et al. [18] further considered heterogeneous user activity levels and real-time content repositories so that they can minimize the average request delay. In [19], Yan et al. designed a UAV-enabled service provisioning framework in extreme environments to obtain a high cache hit ratio and low service delay. Gu et al. [20] presented a coded caching scheme, which is used to minimize the backhaul transmission volume.

Moreover, many researches pay attention to improving energy efficiency through UAV trajectory planning. Liu et al. [21] jointly optimized the UAV's speeds, deployments, and visiting sequences to diminish UAV's mission completion time. Ji et al. [10] designed an alternating method to improve throughput by solving caching decisions, path design and resource distribution iteratively. In [22], Wang et al. introduced a two-layer algorithm named ToDeTaS for the UAVs deployment and mission scheduling optimization collectively. However, due to the high demand for quick decision-making within dynamically changing environments, the aforementioned methods cannot supply comparatively excellent service to users.

Thankfully, DRL is suitable for dynamic changes in UAV-assisted MEC by reducing the solution complexity and has achieved remarkable results in this field. Li et al. [23] designed lightweight low-complexity optimized networks to complete the online trajectory planning. Zhu et al. [24] investigated a UAV-assisted wireless sensor network for the path strategy of UAVs with lower energy consumption. Al-Hilo [25] proposed a proximal policy optimization (PPO) based algorithm for achieving a balance between the data flow offloading and power consumption in an efficient manner. Moreover, since designing a single UAV's trajectory alone but used in multi-UAV environments may lead to service confusion and wasted resources, MARL tackles the multi-agent decision problem in a distributed manner. Ding et al. [26] introduced a probabilistic MADDPG-based algorithm for maximized throughput of ground users and high-efficient communication service with UAV-BSSs. In [17], a MATD3-based approach is designed to optimize joint transmit power and UAV path designs.

Although the above studies are efficient to some extent, there still exist several technical challenges pertinent to UAV-assisted caching in the IoV. First, many researches concentrate on UAV-assisted networks serving low-mobility users, which cannot be directly implemented in an IoV environment. The high-mobility vehicles cause greatly dynamic changes in content request distribution and affect the service performance of UAVs. Nevertheless, due to the constrained storage space and computational power of UAVs, online trajectory design with the caching scheme should be trained under finite computational resources. Moreover, since the service requests and the vehicle network conditions are hard to predict accurately, UAVs should make online decisions to follow the highly dynamic vehicle states, delivering the optimal real-time solution. Our work bridges these gaps through differential evolution and multi-agent proximal policy optimization. This combined framework aims to diminish the system energy consumption, effectively addressing the outlined challenges and enhancing the efficiency of UAV-assisted caching for human-centric consumer applications in IoV.

III. SYSTEM MODEL

This section contains three components. Initially, UAV-enabled content delivery in IoV is introduced. Then, the content cache and energy consumption of the network framework is constructed. Eventually, the energy optimization issue is translated into a MINLP optimization problem. The key notations in this paper are listed in Table I.

TABLE I
KEY NOTATIONS & THEIR DEFINITIONS

Notations	Definitions
\mathcal{N}	The set of UAVs, $\mathcal{N} = \{n = 1, 2, \dots, N\}$
\mathcal{V}	The set of vehicles, $\mathcal{V} = \{v = 1, 2, \dots, V\}$
\mathcal{K}	The set of services, $\mathcal{K} = \{k = 1, 2, \dots, K\}$
d_k	The data size of k
r_k	The result size of k
c_k	The number of instruction to complete k
t	The time interval
H	The height of UAVs
L_n^t	The coordinate of UAV n in t
L_v^t	The coordinate of vehicle v in t
$D_{n,v}^t$	The horizontal distance between UAV n and vehicle v
R_{max}	The maximum horizontal coverage radius of UAV
V_{max}	The maximum speed of UAV
C_n	The cache capacity of UAV

A. System Framework

As depicted in Fig. 1, the UAV-assisted content caching network framework in IoV is designed, which includes a central macro base station (BS), N UAVs and V vehicles. When working in peak hours, the macro BS may be overloaded so cannot meet the requirement of consumer electronics in vehicles. In this situation, the traffic offloading is supported by UAVs quipped with cache storage. We assume that all devices are fitted with an antenna, so as to transmit data and content between others. Multiple UAVs act as flying containers that provide popular content delivery services for nearby vehicles, which can make direct communication without BS.

The consumer electronics in vehicles will transmit various human-centric application requests at the beginning of one period. Firstly, the vehicle asks the nearest UAV for the request. If the UAV is linked with the vehicle and has cached the required content, it will directly transmit the result to the in-vehicle electronics. Otherwise, the request will be sent to the BS for processing. After a certain period, UAVs can learn to dynamically update the caching scheme from the content requests received in the recent period and optimize the flight trajectory to serve the system more efficiently.

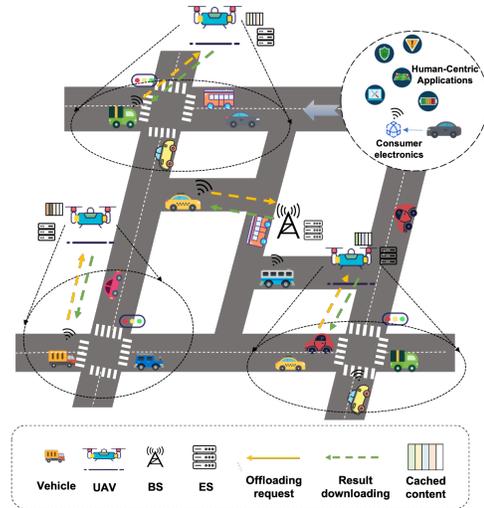


Fig. 1. UAV-assisted content caching network framework in IoV

In this system, it is assumed that the consumer electronics in each vehicle $v \in \mathcal{V}$ generates one or more computationally intensive tasks in each time slot $t \in \mathcal{T}$. Each UAV divides its spectrum resources into multiple wireless channels based on Orthogonal Frequency Division Multiple Access (OFDMA) technology to supply simultaneous channel access for multiple vehicles. User vehicles are provided with wireless access and spectrum access services by accessing neighboring edge nodes. The bandwidth of each channel is smaller than the coherent bandwidth, which does not lead to frequency selective fading of the received signal waveform, i.e., the interference between neighboring channels is negligible. Furthermore, distinct from traditional mobile edge computing networks on the ground, UAV-assisted networks should incorporate UAVs' trajectory planning into the design. Consistent with existing research, 3-D Cartesian coordinates are adopted in this article. Let us define $L_n^t = (X_n^t, Y_n^t, H)$ as the location of the UAV $n \in \mathcal{N}$ in the time interval t . Similarly, vehicle $v \in \mathcal{V}$ travels at a zero altitude and its coordinate can be defined as $L_v^t = (X_v^t, Y_v^t, 0)$. During the mission, UAV $n \in \mathcal{N}$ flies at a fixed height H . Let us define R_{max} as the maximum horizontal coverage radius of UAV. Therefore, the horizontal distance $D_{n,v}^t = \sqrt{\|X_n^t - X_v^t\|^2 + \|Y_n^t - Y_v^t\|^2}$ between UAV n and vehicle v in slot t requires to be within the maximum horizontal coverage radius of UAV R_{max} .

B. Communication Model

In this part, the models for vehicle-to-BS (V2B) communications and UAV-to-vehicle (U2V) communications are introduced as follows.

1) *V2B Communications*: V2B communications can transmit data and service content between the vehicle v and the base station. Only if the vehicle v can't associate with UAV n or there is no cached content it wants on the UAV n , the vehicle v can directly communicate with BS to offload content for processing.

The channel gain between vehicle v and BS is denoted as γ_v , including path loss and a small range of fading, which can be estimated in advance with the help of channel sounding. Hence, the transmission rate can be calculated as

$$o_{vb}^t = B_0 \log_2 \left(1 + \frac{\gamma_v^t p_v}{\sigma^2} \right), \quad (1)$$

where B_0 represents the bandwidth allocated to vehicle v and BS, p_v stands for the transmission power of v and σ^2 is the Gaussian white noise power.

2) *U2V Communications*: Since UAVs may encounter obstacles when transmitting signals with vehicles in cities, which may affect signal transmission effects, the modeling of U2V communications relies on the established shadow model of the line of sight (LoS) link related to the LoS connection probability, depending on the joint environmental conditions and UAV locations. Thus, the LoS connection probability can be calculated by [27]

$$P_r(H, D_{n,v}^t) = \frac{1}{1 + \zeta \exp(-\chi(\arctan(\frac{H}{D_{n,v}^t} - \zeta)))}, \quad (2)$$

where ζ and χ are environment constants depending on the UAV's location. Thus, the average path loss for U2V communication at interval t is [28]

$$\overline{\gamma_{n,v}} = 20 \log \frac{4\pi f_c}{c} + 20 \log \sqrt{H^2 + (D_{n,v}^t)^2} + P_r(H, D_{n,v}^t) \eta_n^{Los} + (1 - P_r(H, D_{n,v}^t)) \eta_n^{NLos}, \quad (3)$$

where f_c and c represent the carrier frequency and the speed of light. η_n^{Los} and η_n^{NLos} are the shadowing variables depending on the environment.

The average SNR of vehicle v from linked UAV n at an interval t is denoted as

$$\varrho_{n,v}^t = \frac{p_v}{\sigma^2}. \quad (4)$$

Furthermore, let γ_{nv}^t be the channel gain between vehicle v and UAV n . Thus, the channel gain can be calculated by

$$\gamma_{nv}^t = \overline{\gamma_{n,v}} D_{n,v}^t{}^{-2} = \frac{\overline{\gamma_{n,v}}}{\|X_n^t - X_v^t\|^2 + \|Y_n^t - Y_v^t\|^2 + H^2}. \quad (5)$$

So the uplink transmission rate from v to n in t can be calculated by

$$o_{vn}^t = B_n \log_2 (1 + \gamma_{vn}^t \cdot \varrho), \quad (6)$$

where B_n is the spectrum bandwidth allocated from vehicle v to UAV n , and ϱ is the average SNR of vehicle v .

C. Caching Model

In order to minimize computation energy, the framework considers that some content files will be cached in UAVs. It is assumed that the consumer electronics in vehicle v traveling on the road requires one or more content services at one-time slot. Then the requirement of vehicle v is calculated by

$$R_k^v(t) = \begin{cases} 1, & v \text{ requires services } k, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

k has three parameters (d_k, r_k, c_k): the data size of k , the result size of k and the computing size of k . Moreover, $\Gamma = \{\eta_k^n\}_{n \in \mathcal{N}, k \in \mathcal{K}}$ is denoted as the cache strategy and η_k^n is denoted as whether the service content k has been cached on the UAV n . If $\eta_k^n = 1$, the content has been cached on the UAV. Otherwise, the content isn't cached. So, for the cached content, we only need to return the corresponding result to the required vehicle. The feedback time of vehicle v can be measured by

$$TCT_v^n(t) = \sum_{k=1}^K R_k^v(t) \cdot \eta_k^n \cdot \frac{r_k}{o_{vn}^t}, \quad (8)$$

In this equation, only when $R_k^v(t) \cdot \eta_k^n$ equals 1, it means that the consumer electronics in vehicle v requires the service content k , and the content is cached on the UAV at the same time. Hence, the computation of vehicle's v energy consumption is expressed as

$$ECT_v^n(t) = \sum_{k=1}^K R_k^v(t) \cdot \eta_k^n \cdot \frac{r_k}{o_{vn}^t} \cdot \psi_n, \quad (9)$$

where ψ_n denotes the offloading power of UAV n . In addition, the amount of energy consumed to cache content at UAV n can be measured by

$$ECC^n(t) = \sum_{k=1}^K \eta_k^n \cdot \frac{c_k}{c_n} \cdot \varpi(n), \quad (10)$$

where $\varpi(n)$ represents the computing power of UAV n .

In addition, considering the limited caching memory size of UAVs, all of the services can't be cached at the same time. Therefore, under the condition of ensuring the popularity of cached service, service contents should be cached as much as possible. The overall storage space taken up by content files shall not exceed the storage capacity C_n of the UAV. Therefore, servers have to properly decide which service to cache. Due to the limited storage space, it should satisfy

$$\sum_{k=1}^K c_k \eta_k \leq C_n. \quad (11)$$

D. Energy Consumption Model

The whole network contains two parts of energy consumption: the moving energy to keep the UAVs in motion and the execution energy for message transmission.

1) *Propulsion Energy*: As stated in [29], for a UAV with speed V_n , the propulsion energy consumption within a time slice t can be calculated by

$$P(V_n) = P_0 \left(1 + \frac{3V_n^2}{U^2} \right) + P_1 \left(\left(1 + \frac{V_n^4}{4v_r^4} \right)^{\frac{1}{2}} - \frac{V_n^2}{2v_r^2} \right)^{\frac{1}{2}} + \frac{1}{2} d_0 \rho s A V_n^3, \quad (12)$$

where P_0 represents the rotor blade profile power of a UAV in a hovering state, P_1 is the rotor blade induced power in hovering, U denotes the tip speed of the blades, v_r signifies the average rotor induced velocity during hover, d_0 is the drag ratio experienced by the UAV, ρ represents air density, s denotes the rotor disc solidity, and A represents the blade disc area. Therefore, the UAV's n propulsion consumption in t is obtained by

$$E_n(t) = P(V_n) \cdot t. \quad (13)$$

2) *Execution Energy*: For cached content on the UAV, only the energy cost of transmitting the processing result and caching content at UAVs is calculated. Otherwise, the non-cached content on UAV needs to experience the following steps: offloading tasks into BS, computing, and transmitting the result. Similar to Eq. 8, the execution time of the content processed on the BS is donated by

$$TO_v^n(t) = \sum_{k=1}^K R_k^v(t) \cdot (1 - \eta_k^n) \cdot \left[\frac{d_k + r_k}{o_{vb}^t} + \frac{c_k}{c_b} \right], \quad (14)$$

where $R_k^v(t) \cdot (1 - \eta_k^n)$ represents the content that which vehicle requires but is not cached on UAV. Thus, the execution energy consumption can be obtained by

$$EO_v^n(t) = \sum_{k=1}^K R_k^v(t) \cdot (1 - \eta_k^n) \cdot \left[\frac{r_k}{o_{vb}^t} \cdot \varpi(b) + \frac{c_k}{c_b} \cdot \psi(b) \right], \quad (15)$$

where $\varpi(b)$ represents the computing power of BS and $\psi(b)$ is the offloading power of BS. Thus, the execution energy consumption of the vehicle v in t is calculated as

$$E_v^n(t) = ECT_v^n(t) + EO_v^n(t). \quad (16)$$

Drawing upon the preceding equations, the overall energy consumption in t is expressed as

$$E_{total}(t) = E_{UAV}(t) + E_{execution}(t) = \sum_{n=1}^N (E_n(t) + ECC^n(t)) + \sum_{v=1}^V E_v^n(t). \quad (17)$$

In Eq. 17, the energy consumption includes UAVs' propulsion power, backhauling the result of cached contents and computing non-cached contents on the BS.

E. Problem Formulation

Considering the above network model, our aim is to decrease the average energy among all UAV-served users at one time period by optimizing content caching policy $\Gamma = \{\eta_k^n\}_{n \in \mathcal{N}, k \in \mathcal{K}}$ and UAVs' flight trajectory $\mathbf{G} :=$

$\{\mathbf{G}_n^t\}_{n \in \mathcal{N}, t \in \mathcal{T}}$. Thus, we formulate content caching problem as

$$\begin{aligned} \min_{\Gamma, \mathbf{G}} \quad & \frac{1}{T} \sum_{t=1}^T E_{total}(t) \\ \text{s.t.} \quad & \sum_{k=1}^K c_k \eta_k \leq C_n, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \\ & \eta_k^n \in \{0, 1\}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \end{aligned} \quad (18)$$

where η_k represents whether the service content k has been cached on the UAV n . Given the discrete nature of content cache variables, the continuous nature of UAV trajectory variables, and the presence of non-convex elements within both the objective function and constraints, this problem belongs to a MINLP problem. In this article, a UAV-assisted content caching method with trajectory design through DE and MAPPO is applied to find the optimal solution.

IV. CTDR DESIGN

This section contains two components. Initially, a cache decision method based on the DE algorithm is proposed with simulated service requests. This dynamic process is updated at the onset of each time period, strategically caching appropriate content from service providers with recent histories of service requirements. Thus, UAVs equipped with these optimal caching schemes can exhibit enhanced responsiveness to consumer electronics requests. Then, a trajectory design method based on MAPPO is trained from the environment so that UAVs can perform air service in an efficient path, resulting in expedited service delivery. Eventually, we form CTDR with the above two methods in Fig. 2.

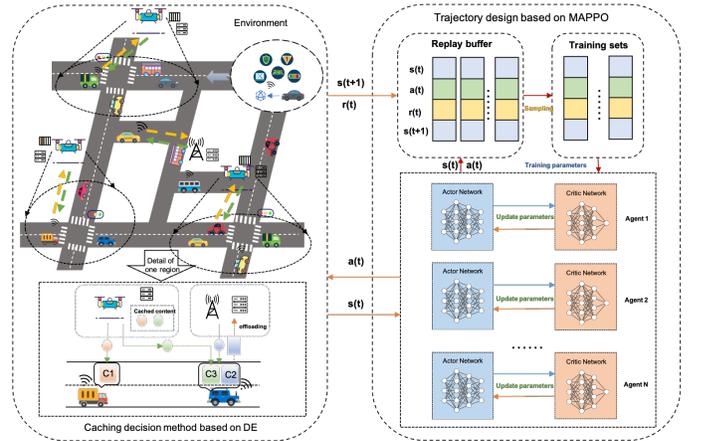


Fig. 2. The framework of energy consumption optimization based on CTDR

A. DE-based Content Caching Scheme

Considering traditional caching decision algorithms limited by some local search or algorithmic complexity, the differential evolution algorithm has strong global search capability within a dynamically changing environment in this paper. It can adapt to nonlinear functions more flexibly, which helps to find a better solution after several iterative optimizations in complex caching scenarios. Therefore, we resort to the

DE algorithm to solve the optimal caching issue. It gives an effective caching scheme and improves cache hit rate $f(\Gamma)$. Following are the specific steps of the DE algorithm.

1) *Population Initialization*: Similar to the traditional genetic algorithm (GA), the decision variables in a differential evolution algorithm are represented as chromosomes. In this paper, the individual consisting of various chromosomes is the edge content caching policy $\Gamma = \{\eta_k^n\}_{n \in \mathcal{N}, k \in \mathcal{K}}$ in our method. Therefore, the content caching scheme can be initialized as

$$\{\Gamma_i(0) | \eta_{k,i,j}^n(0) \in \{0, 1\}; i = 1, 2, \dots, NP; j = 1, 2, \dots, D\}, \quad (19)$$

where $\Gamma_i(0)$ is the i -th initialized individual and the $\eta_{k,i,j}^n(0)$ is the j -dimensional solution vector. NP is the size of populations and $D = N \cdot K$ is solution vectors for one population.

2) *Mutation*: The DE algorithm achieves individual variation through a differential strategy. In this paper, the Best variation selection strategy is proposed [30]. The basic idea is to use the best individual of the current population as one of the different operations, in order to expect to get a better variation vector. The mutation strategy can be donated as

$$M_i(p+1) = \Gamma_{best}(p) + F \cdot (\Gamma_{r1}(p) - \Gamma_{r2}(p)), \quad (20)$$

where $\Gamma_{best}(p)$ is the optimal individual of the population p and F is used to determine the amplification of the deviation variable ratio. Furthermore, the practical significance of $\Gamma_{r1}(p) - \Gamma_{r2}(p)$ is to compute difference vectors. These vectors are instrumental in introducing randomness and diversity, thereby enhancing the exploration of the search space.

3) *Crossover*: Crossover operation is adopted to enhance the diversity within populations. The offspring $U_i(g+1)$ is defined as

$$C_i(p+1) = \begin{cases} M_i(p+1), & \text{if } \text{rand}(0,1) \leq CP \\ \Gamma_i(p), & \text{otherwise} \end{cases} \quad (21)$$

where CP represents crossover probability.

4) *Selection*: The strategy of greedy selection is used in differential evolution, i.e., the better individual $\Gamma_i(g+1)$ selected as the new individual can be calculated by

$$\Gamma_i(p+1) = \begin{cases} U_i(p+1), & \text{if } f(C_i(p+1)) \geq f(\Gamma_i(p)) \\ \Gamma_i(p), & \text{otherwise} \end{cases} \quad (22)$$

After O iterations following the above steps, the optimal caching scheme is obtained from the final individual. Algorithm 1 depicts the detailed content caching scheme. First, the population is initialized before starting model training (line 1). Then mutation (line 4), crossover (line 5), and selection (lines 6-10) operations are performed in sequence for each population individual, respectively. Finally, the optimal caching strategy can be obtained (line 13).

B. MAPPO-based Trajectory Design Optimization

After determining the content cache scheme on each UAV, we should design a trajectory optimization strategy aimed at reducing the whole energy consumption in IoV. Considering the high mobility of UAVs in interacting with vehicles, traditional trajectory planning methods cannot handle the motion

Algorithm 1: DE-based Content Caching Scheme

Input: The size of populations NP , the variable factor in DE F , the crossover probability CP , the number of iterations O

Output: Optimal caching scheme Γ

```

1 Initialize population  $\Gamma_i(0)$ ;
2 for each iteration do
3   for all  $i \in \{1, 2, \dots, NP\}$  do
4     get the mutation strategy  $M_i(p+1)$  by (20);
5     get the crossover offspring  $C_i(p+1)$  by (21);
6     if  $f(C_i(p+1)) \geq f(\Gamma_i(p))$  then
7        $\Gamma_i(p+1) \leftarrow C_i(p+1)$ ;
8     else
9        $\Gamma_i(p+1) \leftarrow \Gamma_i(p)$ ;
10    end
11  end
12 end
13 return  $\Gamma = (\eta_1, \eta_2, \dots, \eta_K)$ 

```

path of UAVs in a timely manner, thereby failing to ensure service effectiveness. Multi-agent reinforcement learning works as an effective method to cope with the dynamics of UAVs' trajectory planning. At the moment, many researches have adopted multi-agent reinforcement learning such as MADDPG [15], [16] and MATD3 [17] in trajectory design and achieved extraordinary results. In the following, we will build the MDP model for UAVs' trajectory design, and then introduce a path-planning trajectory algorithm based on MAPPO.

1) *Definition of MDP model*: For the trajectory design optimization in the IoV network, each UAV is considered as an agent. We establish a MDP to depict the IoV environment as (S, A, P, R) .

- **State**: The state of the environment S includes UAVs' positions, vehicles' locations and the number of task requests for vehicles in each time interval. The state in t is represented as

$$S = \{\mathbb{L}_{uav}(t), \mathbb{L}_{vehicle}(t), F_v(t)\}, \forall v \in \mathcal{V}, t \in \mathcal{T}, \quad (23)$$

where $\mathbb{L}_{uav}(t)$ and $\mathbb{L}_{vehicle}(t)$ represent the set of UAVs' and vehicles' locations in t . $F_v(t)$ donates the number of all task requests during time slot t .

- **Actions**: Each UAV requires should make decisions in accordance with the present state. We define the UAV's deflection angle and speed ratio as the action of each agent, so it is expressed as

$$A = \{\theta_n(t), \xi_n(t)\}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \quad (24)$$

where $\theta_n(t) \in [0, 2\pi]$ is the agent's deflection angle and $\xi_n(t) \in [0, 1]$ represents the agent's speed ratio, which means that the speed of the agent is calculated as $V_n(t) = V_{max} \cdot \xi_n(t)$. Notably, since UAVs are flexible, we build a continuous space.

- **Policies and State Transition Probabilities**: Policy refers to the strategy of an agent to choose actions in different

states, which means a mapping relationship from states to actions. The policy is represented as

$$\pi_{\theta}(a_i|s_i), \quad (25)$$

where θ represents the parameter of the actor networks. After agents take action, the state will transition to a different state. So the state transition probability is obtained by

$$P(s'|s, A), \quad (26)$$

- Rewards: Based on Eq. 18, under the conditions of optimal caching strategy and trajectory planning, the total energy consumption during period T is donated as *Rewards*. Since multi-agent reinforcement learning aims to maximize the rewards, we define the negative value of *Rewards* as

$$R = - \sum_{t=1}^T E^{total}(t). \quad (27)$$

2) *MAPPO-based Trajectory Design Optimization*: As a variant of PPO performance in the field of multi-agent environments, MAPPO has been one of the popular MARL algorithms in recent years. It adopts a centralized value function approach to take into account global information which falls within the centralized training and decentralized execution (CTDE) framework. Thus, a single agent can cooperate with each other through a global value function.

In the off-policy approach, agents' training data originates from historical experience. Since the training data does not depend on the current policy, the off-policy approach may be more advantageous in terms of stability. However, it's noteworthy that in highly dynamic environments, the relevance of past experiences might decrease, possibly impacting learning efficacy. Conversely, the on-policy approach proves suitable for training in such a dynamically changing environment in this paper due to the real-time updating of the policy. The agent will interact with the environment based on the current policy and then use these interaction data to refine the policy. In this context, MAPPO, as an on-policy algorithm, has a significantly higher algorithmic efficiency compared with off-policy MARL algorithms, especially under constrained computational resources [31].

In this paper, UAVs collaborate to provide service for the consumer electronics in vehicles so that all UAVs are assumed to share the parameters of actor and critic networks. Next in each step, the actor and critic network parameters are respectively updated with a policy π_{θ} and a value function $V_{\phi}(s)$. Thus, the loss of actor i is obtained by

$$L(\theta) = \mathbb{E}_{\pi_{\theta}} \min \left[r_{\theta,i}^{(t)} \hat{A}_i^{(t)}, \text{clip}(r_{\theta,i}^{(t)}, 1 - \epsilon, 1 + \epsilon) \hat{A}_i^{(t)} \right], \quad (28)$$

where $r_{\theta,i}^{(t)} = \frac{\pi_{\theta}(a_i^{(t)}|s_i^{(t)})}{\pi_{\theta_{old}}(a_i^{(t)}|s_i^{(t)})}$ is the ratio of the new policy to the old policy, and ϵ is the clipping strength. $\hat{A}_i^{(t)}$ is the generalized advantage estimation (GAE), which is calculated as follows:

$$\hat{A}_i^{(t)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l R_{t+l} - V_{\phi}(s_i^{(t)}), \quad (29)$$

Algorithm 2: MAPPO-based Trajectory Design Optimization

Input: The state of the environment S , learning rate α and β , the number of time intervals T

Output: Strategies for UAV trajectory design

```

1 Initialize replay buffer,  $\theta$ ,  $\phi$  and agents;
2 for each episode do
3   Initialize the amount of task requests and the
   locations of UAVs and vehicles;
4   for each time interval  $t$  do
5     for each UAV  $n$  do
6       The UAV obtains the current state  $s(t)$ ;
7       Choose action  $a_n(t)$  by (25);
8     end
9     The UAVs get their corresponding rewards  $r(t)$ 
   by (27);
10    The environment updates its state  $s(t+1)$ ;
11    Store  $(s(t), a(t), r(t), s(t+1))$  in experience
   replay buffer.
12  end
13  for each training step do
14    Sample some batches of experiences from
   replay buffer;
15    for each UAV  $n$  do
16      Update  $\theta_n$  to decrease  $L(\theta_n)$  by (28);
17      Update  $\phi_n$  to decrease  $L(\phi_n)$  by (30);
18      Update the actor network with learning rate
    $\alpha$  by (31);
19      Update the critic network with learning rate
    $\beta$  by (32);
20    end
21  end
22 end
23 return parameters of network  $\theta$  and  $\phi$ 

```

where γ is the discount factor and λ is the weighting factor of GAE which donates a trade-off between bias and variance [32]. The critic network is learned by using gradient descent with the loss function as follows:

$$L(\phi) = \mathbb{E}_{\pi_{\theta}} \max \left[(V_{\phi}(s_i^{(t)}) - \hat{R}_i)^2, \right. \\ \left. (\text{clip}(V_{\phi}(s_i^{(t)}), V_{\phi_{old}}(s_i^{(t)}) - \epsilon, V_{\phi_{old}}(s_i^{(t)}) + \epsilon))^2 \right], \quad (30)$$

where \hat{R}_i is the estimate of the expected total return for a time interval. Finally, the parameters of networks are updated by

$$\theta \leftarrow \theta + \alpha L(\theta), \quad (31)$$

and

$$\phi \leftarrow \phi + \beta L(\phi), \quad (32)$$

where α and β are the learning rates of actor and critic networks, respectively. Algorithm 2 shows the details of trajectory design method based on MAPPO. At the beginning of each time interval, the amount of content service requests and the locations of UAVs and vehicles are established as the initial state (lines 1-3). Then, each UAV agent will take an action

from the current environment (lines 5-8). Thus, the reward of the environment in this time interval can be computed (line 9). Once the environment incorporates the actions executed by UAVs, the state updates (line 10). The experience tuple of state, actions, reward, and next state are stored within the replay buffer, which can improve the effectiveness of model parameters (line 11). Before training the policy network, the agents first sample some batches of experiences from the replay buffer (line 14). Then, at the end of a training step, the agents will update their policies (lines 15-20).

V. EXPERIMENT AND ANALYSIS

In this part, the CTDR method is presented. We first introduce the experimental environment, the parameters of the simulation environment, and the network parameters of MAPPO. Then, the validity of our proposed caching strategy is evaluated by the cache hit ratio. Next, the convergence performance analysis of trajectory optimization based on MAPPO is conducted. Finally, the proposed caching method is combined with MAPPO to achieve the co-optimization of caching strategy and UAV path planning. Through experiments, the feasibility of the scheme is analyzed from system energy consumption.

A. Experimental Setup

The experiments are conducted on Windows 11 with AMD Ryzen 7 5800H with Radeon Graphics (3.20 GHz, 7 cores), 16.0 GB RAM, and NVIDIA GeForce RTX 3060 Laptop GPU. The Python 3.8, Pytorch 1.10.2 and Numpy 1.20.1 are implemented as the simulation tools. Specifically, in our simulation environment, a $1000 \times 1000 m^2$ area is considered as the target region and vehicles equipped with the consumer electronics are randomly located in this area and traverse at a consistent speed of 10 meters per second along predetermined routes. Similarly, UAVs initiate their flights from service base stations at the beginning of each service period. Each service period T lasts 50 seconds, and one-time interval is assumed to be 1 second. Additionally, the dataset employed for evaluating the CTDR approach, in the context of addressing content caching challenges, is generated based on the simulation outlined in a prior reference [33]. Other parameter settings are shown in Table II, according to [27], [29], [34], [35]. Additionally, the detailed parameters of MAPPO are concluded in Table III. To ensure fairness, the hyperparameters for MADDPG and MATD3 training are set to the same, with a unique parameter for soft updating the target network of MADDPG and MATD3 set to 0.01.

B. Comparative Analysis of DE-based Content Caching Scheme

In this part, we generate 15 different types of content caching in UAV. It is assumed that the total number of requests per second is different. To analyze the performance of the DE-based scheme, four content caching schemes are proposed as follows.

- Size first caching scheme (SFC): Choose smaller contents to cache first.

TABLE II
PARAMETER SETTINGS

Parameters	Values
H	100 m
V	10
R_{max}	200 m
V_{max}	20 m/s
ζ, χ	11.9, 0.13
P_0, P_1	79.86 W, 88.63 W
$\eta_n^{Los}, \eta_n^{NLos}$	1.6, 23
ψ_n	3 W
$\varpi(n)$	250 W
$\varpi(b)$	300 W
p_v	100 W
c_b	2500 MIPS
d_k, r_k, c_k	10 to 50 MB, 3 to 10 MB, 300 to 3500 million instructions, randomly distributed

TABLE III
PARAMETER OF MAPPO MODEL

Parameter	Value
Discount factor of rewards	0.99
Episode length	50
Learning rate of actor networks	0.001
Learning rate of critic networks	0.001
Number of hidden layers	3
Dimension of hidden layers	64
Number of PPO epochs	10
Number of mini-batches	1
PPO clip parameter	0.2
Optimizer	Adam

- Random caching scheme (RC): Randomly choose contents to cache.
- Popularity first caching scheme (PFC): Choose more popular contents to cache first.
- SGA-based content caching scheme (SGA): Choose appropriate contents to cache based on the simple GA.

In this part, the hit ratio can be expressed as performance indicators for the proposed algorithm and the comparison algorithm, which can be obtained by

$$\Phi = \frac{\sum_{k=1}^K R_k^v(t) \cdot \eta_k^n}{\sum_{k=1}^K R_k^v(t)} \quad (33)$$

In the experiment, the caching capacity of the UAV is set to 20, 40, 60, 80 and 100 MB. The experimental results of the hit rate of these five methods are shown in Fig. 3. As the caching capacity of a UAV increases, the hit ratio of each method is increased to varying degrees. However, our method always performs better compared with the other 4 algorithms. Specifically, when the caching capacity is limited to 20 and 40 MB, SFC is greatly affected, with only 6.87% and 12.96% cache hit ratio. However, our algorithm shows significant performance which is respectively 1.75–2.34 times that of PFC and RC. Thus, the algorithm proposed in this article can better cope with the caching decision problem in the resource-limited case. When the caching capacity reaches 60 MB, our method performances 21.98% and 63.52% better than PFC and RC, respectively. When the caching capacity is above 80 MB, our algorithm still shows advantages to RC

and PFC. Moreover, when compared with SGA, our method always performs the best or equal to the baseline which has a 0.91% – 23.66% improvement. This clearly demonstrates that traditional cache decision-making algorithms are limited by algorithmic complexity. In contrast, the algorithm proposed in this paper possesses a robust global search capability, particularly suitable for dynamically changing environments, realizing a better improvement in computational efficiency. Therefore, the cache hit rate comparative outcomes among the five schemes prove CTDR's superiority.

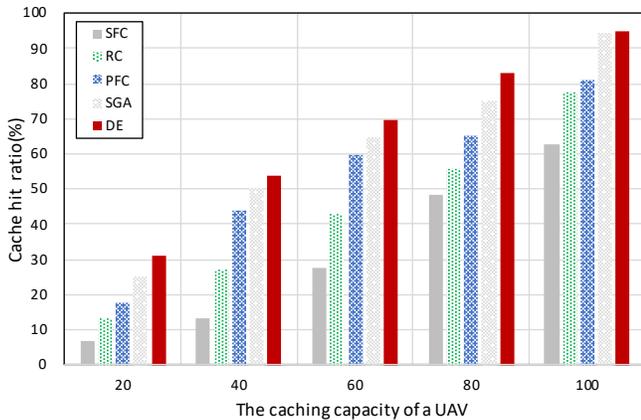


Fig. 3. Comparison on cache hit ratio with SFC, RC, PFC, SGA and our method

C. Convergence Performance Analysis of Trajectory Optimization Based on MAPPO

In this section, we temporarily choose a stable content caching decision to assist UAV's service for vehicles. To evaluate the convergence performance, two hyperparameters (learning rate and PPO clipping strength) are adjusted to different values.

1) *Analysis of the impact of learning rate on performance:* The learning rate determines the extent of each gradient descent update. A high learning rate leads to large update steps, which may lead to non-convergence or even oscillation of the algorithm, while a small learning rate leads to slow convergence and takes a longer time to reach a local optimum. In this paper, it is assumed that the learning rates of actor and critic networks change with the same value. Different learning rates experimented on in our method are shown in Fig. 4. When the learning rate reaches 0.005, due to the large update steps, MAPPO finally converges to a poor result. When the learning rate is 0.0005 or 0.0008, the method reaches a local optimum at a slow speed. However, when it gets 0.001, the algorithm converges to a higher average reward, and the convergence speed is relatively fast.

2) *Analysis of the impact of PPO clipping strengths on performance:* Clipping strength is another core hyperparameter that significantly affects convergence performance: a large clipping strength leads to too large update steps, which makes the update process unstable and may cause the model to converge to a suboptimal solution, while a small clipping strength

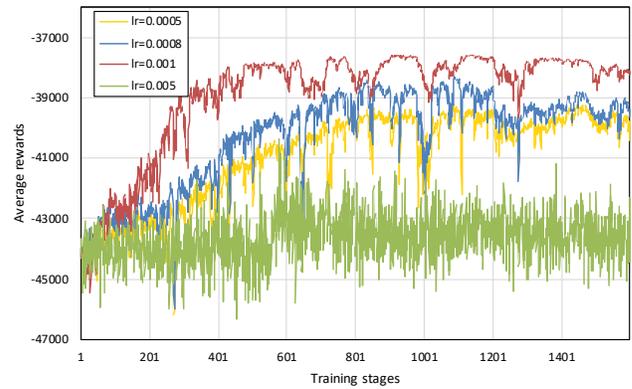


Fig. 4. Average rewards under different learning rates

may limit updates to new policies and lead to slow convergence. In Fig. 5, different clipping strengths are experimented with under the fixed learning rate of 0.001. When the clipping strength reaches 0.1 or 0.15, it leads to non-convergence or even oscillation of the method and similar training curves. When the clipping strength is 0.2, MAPPO obtains a relatively effective and stable performance. Generally, clipping strength which reaches 0.2 has an around 5% improvement over the other values.

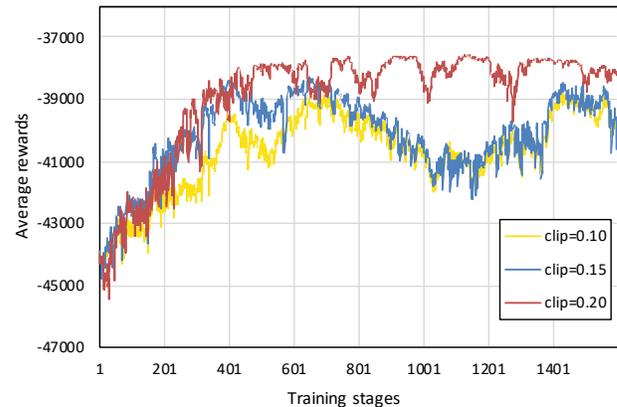


Fig. 5. Average rewards under different PPO clipping strengths

D. Comparative Analysis of CTDR

In this section, the following three methods are applied for performance comparison to prove the feasibility of our method, and these schemes are described as follows:

- MADDPG-based trajectory design with DE caching scheme (DEDT): Use MADDPG for optimal trajectory design with DE caching scheme, which performs well in the continuous control field [16].
- MATD3-based trajectory design with DE caching scheme (DETT): Use MATD3 to solve the energy consumption problem, which is another off-policy MARL as the optimization of MADDPG.

- MAPPO-based trajectory design with a random caching scheme (RCPT): Use a random caching scheme to evaluate the influence of caching policy on energy consumption.

Since the reward defined at MAPPO is the negative value of the energy consumption, we donate the average consumption in a time period as the evaluation metrics to illustrate the performance of CTDR:

$$E_{ave} = \frac{1}{T} \sum_{t=1}^T E_{total}(t) \quad (34)$$

The comparison of energy consumption is depicted in Fig. 6. It is evident that with the expansion of UAV caching capacity, the energy consumption of each method is continuously reduced. Moreover, on the fixed size of caching capacity, CTDR and RCPT apart from 20 MB always perform better than other methods, which proves that MAPPO has a higher algorithmic efficiency than MADDPG and MATD3 in our system environment. For the energy consumption of one training step, our method reduces 4.93% – 11.79% compared to DEDT and 3.85% – 7.05% compared to DETT. In addition, when the caching capacity is set to 20 or 40 MB, the energy consumption optimization by our method is about 4.79% – 7.77% lower than by RCPT, which proves the significant impact of caching policy on energy consumption. As the caching capacity of a UAV expands from 60 to 100 MB, the variance between CTDR and RCPT gradually diminishes, which verifies the impact of distinct caching methods on energy consumption becomes less pronounced as the cache space attains a sufficient scale. Thus, it is evident that CTDR can better optimize the system's energy consumption.

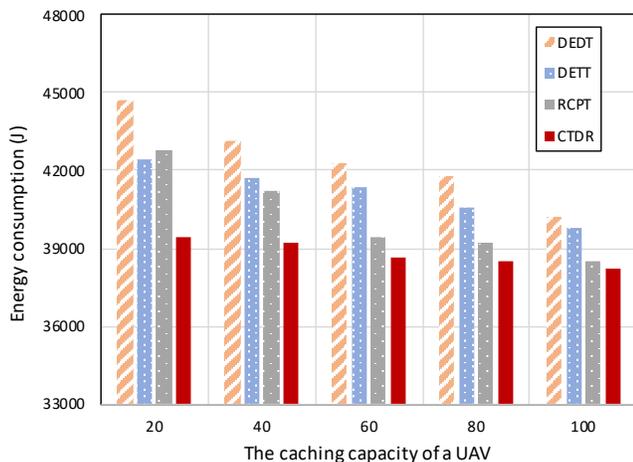


Fig. 6. Comparison on energy consumption under different caching capacities of a UAV among DEDT, DETT, RCPT and CTDR

Fig. 7 elaborates the number of requests per second also influences energy consumption. Evidently, as the demand requests escalate, the energy consumption of the four schemes experiences a corresponding increase. This phenomenon can be attributed to the inherent challenge faced by UAV-enhanced edge computing in accommodating computationally intensive

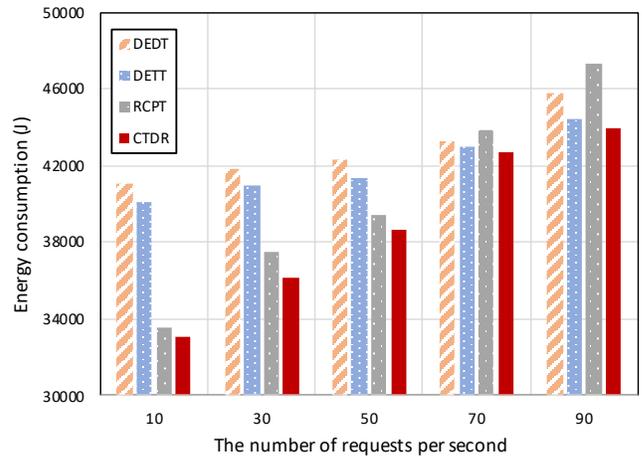


Fig. 7. Comparison on energy consumption under different numbers of requests per second among DEDT, DETT, RCPT and CTDR

vehicle services that necessitate a greater allocation of processing resources for execution. However, compared to other methods, our method always consumes the lowest energy. As the number of requests increases, DEDT and DETT maintain a respectively low growth rate compared to RCPT and CTDR, which proves the efficiency of off-policy MARL algorithms in large-scale systems. However, when the number of requests is limited under 30, CTDR reduces 11.83% – 19.59% energy consumption more than DEDT and DETT, which shows that MAPPO has better generalizability with fewer samples. Furthermore, the CTDR approach exhibits superior energy consumption performance in comparison to the RCPT approach. This reason can be attributed to the effectiveness of the DE-based caching scheme, which outperform the relatively simplistic approach of random caching employed by RCPT.

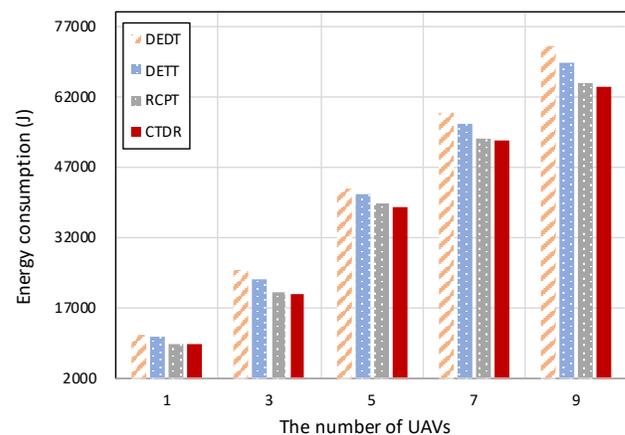


Fig. 8. Comparison on energy consumption under different numbers of UAVs among DEDT, DETT, RCPT and CTDR

Fig. 8 portrays the energy consumption of different numbers of UAVs under these four methods. From Fig. 8, it is evident that the amount of UAVs used in IoV becomes the major

significant factor in energy consumption by comparison with other factors. Besides, as the number of UAVs increased from 5 to 9, the variance in the total energy consumption between the two gradually stabilizes. This phenomenon is due to the fact that when the volume of service requests remains consistent, most of the service requirements can be satisfied within 5 UAVs. Therefore, the difference is mainly attributed to excess energy consumption from the energy consumption of excessive UAVs, leading to an ineffective utilization of resources. For the off-policy algorithms employed in comparative experiments, the agents' training data is derived from historical experiences. In highly dynamic environments, the relevance of past experiences may diminish, potentially impacting computational efficiency. Based on the results presented above, it is evident that CTDR exhibits significantly superior algorithmic efficiency, particularly when computational resources are limited.

The significant advantage of employing CTDR for simulating joint content caching and trajectory planning strategies resides in its aptitude to accurately replicate the real-world UAV-assisted caching decision-making process in virtual environments under constrained computational, based on simulated data and physical models. By doing so, CTDR effectively decreases potential costs to consumers in real scenarios caused by deploying low-performance UAV-enhanced service in IoV. This approach ensures that real-world scenarios remain unaffected, while CTDR facilitates thorough exploration and analysis, safeguarding against suboptimal deployment decisions. Overall, our method consistently exhibits lower energy consumption compared to alternative approaches, signifying its effectiveness in energy optimization within the IoV landscape.

VI. CONCLUSION

Responding to the challenge in UAV-assisted content caching for human-centric applications in IoV, a cache-enabled consumer electronics network framework was designed. To diminish the system's energy consumption, a content caching algorithm with trajectory design and deep reinforcement learning, named CTDR, was employed. This paper concluded with several simulation experiments, which verified the superiority of the CTDR method in energy consumption optimization compared with other algorithms. This validation underscores its enhanced applicability for human-centric consumer applications within the IoV realm. Nevertheless, it is pivotal to acknowledge that the conclusions above are contingent upon certain assumptions. Specifically, these findings underpin the assumption of uniform communication resource allocation and the UAVs functioning only as cache units instead of edge nodes for data offloading. When confronted with traffic congestion, this model could encounter limitations and its efficacy could diminish. In the future, we will concentrate on optimizing resource efficiency in IoV by considering dynamic resource allocation and distributed offloading schemes for consumer electronics in vehicles.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China under Grant (No. 62372242

and 92267104), the Natural Science Foundation of Jiangsu Province of China under Grant (No. BK20211284), the National Student Innovation and Entrepreneurship Training Program Support Project (No. 202310300038Z) and the NUIST Students' Platform for Innovation and Entrepreneurship Training Program (No. XJDC202310300353).

REFERENCES

- [1] B. Markwalter, "The consumer technology market-past and present [cta insights]," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 108–C3, 2019.
- [2] C. K. Wu, C.-T. Cheng, Y. Uwate, G. Chen, S. Mumtaz, and K. F. Tsang, "State-of-the-art and research opportunities for next-generation consumer electronics," *IEEE Transactions on Consumer Electronics*, 2022.
- [3] X. Xu, Q. Jiang, P. Zhang, X. Cao, M. R. Khosravi, L. T. Alex, L. Qi, and W. Dou, "Game theory for distributed iov task offloading with fuzzy neural network in edge computing," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 11, pp. 4593–4604, 2022.
- [4] X. Xue, G. Li, D. Zhou, Y. Zhang, L. Zhang, Y. Zhao, Z. Feng, L. Cui, Z. Zhou, X. Sun, *et al.*, "Research roadmap of service ecosystems: A crowd intelligence perspective," *International Journal of Crowd Science*, vol. 6, no. 4, pp. 195–222, 2022.
- [5] X. Xue, X. Yu, D. Zhou, C. Peng, X. Wang, Z. Zhou, and F. Wang, "Computational experiments: Past, present and perspective," *Acta Automatica Sinica*, vol. 49, no. 2, pp. 1–26, 2023.
- [6] W. Liu, X. Xu, L. Wu, L. Qi, A. Jolfaei, W. Ding, and M. R. Khosravi, "Intrusion detection for maritime transportation systems with batch federated aggregation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2503–2514, 2022.
- [7] H. Yan, M. Bilal, X. Xu, and S. Vimal, "Edge server deployment for health monitoring with reinforcement learning in internet of medical things," *IEEE Transactions on Computational Social Systems*, 2022.
- [8] C. Dong, Y. Shen, Y. Qu, K. Wang, J. Zheng, Q. Wu, and F. Wu, "Uavs as an intelligent service: Boosting edge intelligence for air-ground integrated networks," *IEEE Network*, vol. 35, no. 4, pp. 167–175, 2021.
- [9] Y. Qu, H. Dai, H. Wang, C. Dong, F. Wu, S. Guo, and Q. Wu, "Service provisioning for uav-enabled mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3287–3305, 2021.
- [10] J. Ji, K. Zhu, D. Niyato, and R. Wang, "Joint cache placement, flight trajectory, and transmission power optimization for multi-uav assisted wireless networks," *IEEE Transactions on wireless communications*, vol. 19, no. 8, pp. 5389–5403, 2020.
- [11] R. Liu, A. Liu, Z. Qu, and N. N. Xiong, "An uav-enabled intelligent connected transportation system with 6g communications for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [12] H. Yan, X. Xu, F. Dai, L. Qi, X. Zhang, and W. Dou, "Service caching for meteorological emergency decision-making in cloud-edge computing," in *2022 IEEE International Conference on Web Services (ICWS)*, pp. 120–128, IEEE, 2022.
- [13] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and ai for uavs: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15435–15459, 2022.
- [14] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, and J.-H. Kim, "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-uav control," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7086–7096, 2022.
- [15] Y. He, Y. Gan, H. Cui, and M. Guizani, "Fairness-based 3d multi-uav trajectory optimization in multi-uav-assisted mec system," *IEEE Internet of Things Journal*, 2023.
- [16] A. Gao, Q. Wang, W. Liang, and Z. Ding, "Game combined multi-agent reinforcement learning approach for uav assisted offloading," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12888–12901, 2021.
- [17] S. Zhou, Y. Cheng, X. Lei, Q. Peng, J. Wang, and S. Li, "Resource allocation in uav-assisted networks: A clustering-aided reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12088–12103, 2022.
- [18] J. Luo, J. Song, F.-C. Zheng, L. Gao, and T. Wang, "User-centric uav deployment and content placement in cache-enabled multi-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5656–5660, 2022.

- [19] H. Yan, H. Li, X. Xu, and M. Bilal, "Uav-enhanced service caching for iot systems in extreme environments," *IEEE Internet of Things Journal*, 2023.
- [20] S. Gu, X. Sun, Z. Yang, T. Huang, W. Xiang, and K. Yu, "Energy-aware coded caching strategy design with resource optimization for satellite-uav-vehicle-integrated networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5799–5811, 2021.
- [21] K. Liu and J. Zheng, "Uav trajectory optimization for time-constrained data collection in uav-enabled environmental monitoring systems," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24300–24314, 2022.
- [22] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-uav-enabled mobile edge computing," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2019.
- [23] S. Li, F. Wu, S. Luo, Z. Fan, J. Chen, and S. Fu, "Dynamic online trajectory planning for a uav-enabled data collection system," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13332–13343, 2022.
- [24] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "Uav trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9540–9554, 2021.
- [25] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "Uav-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5155–5167, 2020.
- [26] R. Ding, Y. Xu, F. Gao, and X. Shen, "Trajectory design and access control for air-ground coordinated communications system with multi-agent deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5785–5798, 2021.
- [27] S. Anokye, D. Ayepah-Mensah, A. M. Seid, G. O. Boateng, and G. Sun, "Deep reinforcement learning-based mobility-aware uav content caching and placement in mobile edge networks," *IEEE Systems Journal*, vol. 16, no. 1, pp. 275–286, 2021.
- [28] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.
- [29] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for uav-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3536–3550, 2021.
- [30] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm and evolutionary computation*, vol. 44, pp. 546–558, 2019.
- [31] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24611–24624, 2022.
- [32] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, "Energy efficient computation offloading in aerial edge networks with multi-agent cooperation," *IEEE Transactions on Wireless Communications*, 2023.
- [33] X. Xu, Z. Fang, J. Zhang, Q. He, D. Yu, L. Qi, and W. Dou, "Edge content caching with deep spatiotemporal residual network for iov in smart city," *ACM Trans. Sen. Netw.*, vol. 17, jun 2021.
- [34] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.
- [35] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.



Researcher of Clarivate 2021 and 2022. His research interests include edge intelligence and service computing.



is currently working as a Senior Lecturer with School of Computing and Communications. His research interests include design and analysis of network protocols, network architecture, network security, the IoT, named data networking, blockchain, cryptography, and future Internet.



China. He has multi-disciplinary expertise and working experience on diverse topics of Big Data Analytics, Cloud Computing, Predictive Maintenance, Explainable AI, Knowledge Graphs, Data Science and machine learning. He has more than 10-years of professional experience while working both in industry and academia. He worked in multinational companies like Siemens and Atos. Dr. Khan is a certified Google Cloud Professional Architect. He is an active researcher working on various projects with multiple collaborators and currently he is working on European Union project titled "Human-AI Teaming Platform for Maintaining and Evolving AI Systems in Manufacturing". He also won a project during Pakistan Scientific Foundation (PSF) CRP4 call as Principal Investigator.



Wen Wang is currently pursuing the B.S. degree in software engineering with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include mobile edge computing and deep reinforcement learning.



Yizhou Xing is currently pursuing the B.S. degree in Big Data with the Reading Academy, Nanjing University of Information Science and Technology. Her research interests include software engineering and big data analysis mining and processing.