Joint Deterministic and Probabilistic Edge Caching Minimized Service Time for Cooperative Video Transmission in VANETs

Quynh-Anh Nguyen, Nguyen-Son Vo, Thuong C. Lam, Tan Do-Duy, Haejoon Jung, *Senior Member, IEEE* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—Video streaming in vehicular ad-hoc networks (VANETs) faces significant challenges due to the dynamic nature of vehicles, frequent disconnections, and huge demands for high data rate communications. These challenges make it longer for vehicle users (VUs) to complete their sessions in video applications and services (VASs). In this paper, we fully utilize the benefits of both deterministic and probabilistic edge caching (DPC) techniques for cooperative transmission to minimize the service time in VASs. To do so, a DPC optimization problem is formulated and solved for the optimal results of 1) caching placement in roadside units (RUs) and 2) caching probability in VUs under the constraint on caching storage resource. Genetic algorithms are modified to deal with the complexity of two types of optimization variables, i.e., integer variable for deterministic caching and real variable for probabilistic caching, and thus ensuring high stability and accuracy. Simulation results demonstrate that the DPC method outperforms the other conventional schemes in terms of service time while efficiently utilizing the storage of RUs and VUs. Important findings are also analyzed and discussed to provide more useful insights into the design of edge caching techniques for VASs in VANETs.**

***Keywords*—Caching networks, deterministic caching, probabilistic caching, vehicular ad-hoc networks, video streaming applications and services.**

I. Introduction

Vehicular ad-hoc networks (VANETs) have become an indispensable component of modern intelligent transportation systems, enabling various communication applications and services among vehicle users (VUs) and roadside infrastructure in sixth-generation (6G) networks [1]. Consequently, automotive manufacturers have developed a wide range of in-vehicle applications and services, e.g., infotainment, automated driving, over-the-air update, and advanced driver assistance [2]. The crucial role of in-vehicle applications and services triggers the proliferation of data dissemination in VANETs, resulting in a substantial backhaul workload for macro base stations (MBSs), small-cell base stations, and roadside units (RUs) [3]–[7]. In VANETs, it is more challenging due to the movement of vehicles, the frequent disconnections, and especially the huge demands for high data rate amongst delay-sensitive communications in video applications and services (VASs). To address these challenges, instead of deploying costly high-speed backhaul links, software-defined edge techniques can be efficiently adopted as an alternative [8].

One of the most promising software-defined edge techniques that has attracted considerable attention in 6G VANETs research is caching [9]–[11]. In the 6G era, driven by the disruptive advancements in big data, data science, high-performance computing, and artificial intelligence, prediction models for caching have become increasingly accurate, thereby enhancing the overall performance of VANETs. A significant benefit of caching is that it provides high resource efficiency and reduces infrastructure investment costs for Internet service providers and content providers, while also improving the quality of experience for mobile users. To this end, emerging caching methods have been proposed for both terrestrial and dynamic air networks — ranging from VANETs to content delivery, device-to-device, Internet of Things, unmanned aerial vehicles, and satellite networks [8], [12]–[16].

In fact, caching in VANETs differs from other networks due to diverse factors of applications and services, complex characteristics of VUs, highly dynamic environments, and non-ubiquitous roadside infrastructure [7], [17]. Therefore, to enhance the performance of VANETs, caching involves various contents, tasks, and services (CTSs), each associated with specific popularity patterns. It also accounts for the mobility, distribution, social relationships, incentives, and available resources of VUs. Caching is categorized into deterministic and probabilistic methods. These methods can be combined and then assisted by other techniques (e.g., clustering [18], [19], cooperative transmission (CoT) [4]–[6], [20]–[22], and broadcast transmission scheduling (TrS) and transmission rate adaptation (TrA) [23]) to address the highly dynamic environ-

Quynh-Anh Nguyen is with Vietnam Aviation Academy, Ho Chi Minh City 70000, Vietnam (e-mail: anhnq@vaa.edu.vn).

Nguyen-Son Vo is with the Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, 70000, Vietnam, and also with the Faculty of Electrical-Electronic Engineering, Duy Tan University, Da Nang, 50000, Vietnam (e-mail: vonguyenson@duytan.edu.vn).

Thuong C. Lam is with the HUTECH Institute of Engineering, HUTECH University, Ho Chi Minh City 70000, Vietnam (e-mail: lc.thuong@hutech.edu.vn).

Tan Do-Duy is with Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, 70000, Vietnam (e-mail: tandd@hcmute.edu.vn).

Haejoon Jung is with Kyung Hee University, Yongin, Gyeonggido, Republic of Korea (e-mail: haejoonjung@khu.ac.kr).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada and also with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, U.K (e-mail: tduong@mun.ca).

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2023.25.

Corresponding author is Nguyen-Son Vo (vonguyenson@duytan.edu.vn).

ments and non-ubiquitous roadside infrastructure.

By caching, the frequently accessed CTSs are stored in the edge of VANETs, closer to the VUs, to facilitate the delivery of in-vehicle applications and services. This approach can relax the workload on the backhaul links through offloading, improve the quality of service (QoS) by shortening the transmission distances, and provide high accessibility to more cached CTSs. It also enables the efficient use of energy, spectrum, and storage resources. In deterministic caching (DC) methods, the CTSs are placed in stationary RUs [19] to leverage the infrastructure-to-vehicle communications and reliable storage resources along the roads [4]–[6], [19], [20], [23]–[26]. This approach enables the pre-planned and systematic optimization of content placement. However, limited roadside infrastructure — mainly due to costly deployment [7], hinders seamless applications and services, particularly for high data rate and delay-sensitive VASs requested by VUs in dense VANETs. By contrast, probabilistic caching (PC) allows caching in the dynamic VUs [22], [27]–[31]. Here, because the VUs are dynamic and intermittently connected while moving together for a certain duration, the PC strategy is widely used to adapt to the mobility patterns of VUs. More importantly, the PC methods can effectively compensate for the limitations of DC methods — the higher density of VUs offers more caching opportunities, thereby enabling more seamless applications and services, even when the VUs join or leave randomly [13].

To elaborate on the advantages and disadvantages of the DC and PC methods, a comparative summary of their key features is presented in Table I. In particular, it is certain that most methods utilize the mobility of VUs and the popularity of CTSs for optimally caching in the MBS, RUs, or/and VUs. Some of them further consider other techniques (i.e., clustering, CoT, TrS, and TrA) to improve the caching performance. However, most of these studies relied on a small number of RUs, which is impractical. Furthermore, although optimization algorithms and solutions have been proposed, their convergence, stability, and accuracy have not been thoroughly investigated. More importantly, the caching methods are deployed separately, which prevents them from fully exploiting the combined benefits of DC and PC to enhance the performance of applications and services in VANETs.

In this paper, motivated by the aforementioned discussions, we propose a joint deterministic and probabilistic edge caching (DPC) method that fully leverages DC at the RUs and PC at the VUs for cooperative video transmission in VANETs. The objective is to minimize the service time and utilize the caching storage resource. The main contributions of this paper are summarized as follows:

- We propose a DPC model for VASs in VANETs. The DPC model exploits the storage resources in both RUs and VUs to enable a cooperative video transmission via RU-to-VU (R2V) and VU-to-VU (V2V) communications. This approach reduces the service time for the VUs who simultaneously access the VASs while driving.
- In DPC model, the DC and PC methods are applied to the stationary RUs and the dynamic VUs. The DPC optimization problem is formulated to find the optimal results of 1) where to cache the video contents in the RUs and 2) caching probabilities of video contents in the VUs. For more practical purposes, we take into account various aspects of VASs in VANETs including the mobility of VUs, the popularity of videos, a large number of RUs, and the CoT technique.
- Due to the complexity of optimization variables, GA is modified to solve the DPC optimization problem with respect to (w.r.t) integer optimization variable for DC and real optimization variable for PC. The convergence, stability, and accuracy of the GA in the DPC optimization problem are also provided.
- Simulation results are presented to demonstrate the feasibility of GA and the benefits of DPC method compared to other schemes. Detailed analyses and discussions are provided with interesting findings and more useful insights into designing edge caching techniques for VASs in VANETs.

The rest of this paper is organized as follows. In Section II, we elaborate on the related work of caching in VANETs. Section III introduces the system model for VASs in VANETs, describes how it works, and present the formulations to derive the objective function of the DPC optimization problem. The DPC optimization problem is formulated and solved by GA in Section IV. Section V provides simulation results and findings to evaluate and demonstrate the benefits of the proposed DPC method compared to other schemes. Finally, we conclude the paper in Section VI.

II. Related Work

In this section, we elaborate on the related work of caching methods in VANETs, as summarized in Table I. First, cache-enabled applications and services in VANETs are introduced to highlight the role of caching in improving the QoS and the system resource utilization. Then, the DC and PC methods are examined to identify the existing limitations, reinforce the motivation, and clarify the contributions of our work.

A. Cache-enabled Applications and Services in VANETs

When the resources in VANETs are fully leveraged, vehicle-as-a-service can emerge as a platform to offer diverse applications and services to the VUs, particularly in smart cities [1]. By utilizing the storage of VANETs, caching is designed as a means to facilitate the delivery of massive data generated by applications and services such as public safety, advanced driver assistance, autonomous driving, over-the-air update, handover management, infotainment, digital twins (DT), and even emerging metaverse [5], [9], [38]–[45]. Each of the applications and services has its own appropriate role of caching, which is described in detail below.

For public safety and advanced driver assistance in VANETs [1], [38], responses to frequent safety instruction requests can be cached as much as possible in the VUs to reduce the workload on the backhaul links. In autonomous driving with over-the-air update [2], [5], [9], [39], caching is integrated with edge computing to offload the highly reusable contents, services, and computation-intensive tasks to the edge servers.

TABLE I. FEATURES OF DC AND PC CACHING METHODS.

Features / Ref. No.	DC	PC	Caching placement	Mobility	Popularity	No. of RUs	Convergence	Stability	Accuracy	Assisted technique
[24]	Yes	No	RUs	Yes	Yes	3	N/A	N/A	Fairly good	No
[26]	Yes	No	RUs	Yes	Yes	6	N/A	N/A	N/A	No
[18]	Yes	No	RUs	Yes	Yes	10	Fairly good	Fairly	Fairly good	Clustering
[19]	Yes	No	RUs	Yes	Yes	30	N/A	N/A	N/A	Clustering
[20]	Yes	No	RUs	Yes	Yes	10	N/A	N/A	N/A	CoT
[23]	Yes	No	RUs	Yes	Yes	7	N/A	N/A	N/A	TrS, TrA
[4]	Yes	No	RUs, MBS	Yes	Yes	2	Good	N/A	N/A	CoT
[5], [6]	Yes	No	RUs, MBS	Yes	Yes	30, 2	Fairly good	N/A	Fairly good	CoT
[25]	Yes	No	RUs, MBS	Yes	Yes	10	Fairly good	N/A	Fairly good	No
[21]	Yes	No	RUs, VUs	Yes	Yes	4	N/A	N/A	N/A	CoT
[32]	Yes	No	RUs, VUs	No	Yes	4	Fairly good	N/A	N/A	CoT
[33]	Yes	No	RUs, VUs	Yes	Yes	2	N/A	N/A	Fairly good	Clustering
[34]	Yes	No	RUs, VUs	Yes	Yes	6	N/A	N/A	N/A	No
[35]	Yes	No	VUs	Yes	Yes	0	N/A	N/A	N/A	No
[36]	Yes	No	VUs	Yes	Yes	0	Fairly good	N/A	Fairly good	No
[37]	Yes	No	VUs	Yes	Yes	0	Fairly good	N/A	Fairly good	No
[27]	No	Yes	VUs	Yes	Yes	0	N/A	N/A	N/A	No
[28]	No	Yes	VUs	Yes	Yes	0	N/A	N/A	N/A	No
[22]	No	Yes	VUs	Yes	Yes	0	N/A	N/A	N/A	CoT
[31]	No	Yes	VUs	Yes	Yes	30	N/A	N/A	N/A	No
[30]	No	Yes	VUs	Yes	Yes	0	N/A	N/A	N/A	No
[29]	No	Yes	VUs	Yes	No	0	N/A	N/A	N/A	No
Our proposed DPC	Yes	Yes	RUs, VUs	Yes	Yes	25	Good	Good	Good	CoT

This approach improves both the quality of experience and the functionality of connected autonomous vehicular systems. For infotainment in VASs, caching is deployed in the VUs to assist handover management with minimum cache size, while maximizing the average rate to ensure continuous transmission [40]. To ensure high-quality video streaming over time-varying VANETs, a video can be divided into multiple quality versions and cached in the MBS for adaptive delivery to the VUs [41].

Regarding DT applications, due to massive content and service delivery, caching in VANETs plays a powerful role in facilitating the delay-bounded content transmission and improving the hit rate. The cache-assisted DT applications can model the inter-vehicle social dynamics, anticipate the vehicular and service request flow patterns, and intelligently orchestrate the communications and storage resources to ensure continuous and efficient content delivery [42]–[44]. In the context of metaverse-driven autonomous vehicles and intelligent transportation systems, caching becomes increasingly complex, requiring careful decisions on what, where, when, and how to cache in order to achieve high hit rate and successful transmission [45].

B. DC Methods

Widely studied in the literature, DC methods are considered as a core strategy for building a caching platform to deliver applications and services in VANETs [4]–[6], [18]–[21], [23]–[26], [32]–[37]. In [24], the authors introduced an mobility prediction module that can estimate the connections between the VUs and the RUs in advance. Then, a learning-based algorithm is applied to find which popular contents to cache and when and where to cache them in the RUs for optimal resource utilization, especially reducing the bandwidth consumption. Similarly, by predicting the mobility of VUs and the popularity of contents, the authors in [26] proposed a flexible method that can enable the RUs to cooperate with each other for caching. Due to the complexity of caching optimization problem, practical algorithms are developed into the form of knapsack problem and suboptimal relaxations for close-to-optimal caching decision.

Caching in the RUs becomes considerably more effective when assisted by clustering techniques [18], [19]. In [18], the VUs with similar trajectory behavior are clustered for caching. The proposed approach formulates a delay-aware clustering-based caching optimization problem and solves it by using a multi-agent deep Q-network (DQN), aiming to minimize energy consumption while ensuring low delay. As shown in [19], clustering the RUs based on the mobility pattern of VUs, download deadline, and storage resources leads to improved caching efficiency. By employing a Markov clustering-based maximum visit probability algorithm for joint optimal caching and clustering solution, the hit ratio is significantly increased while satisfying download deadline constraint.

The performance of caching in the RUs can be improved by the CoT technique between VU-MBS mode and VU-RU mode as studied in [20]. It means that the decision to cache in the RUs depends on the channel quality under both modes, aiming to achieve high hit ratio and low delay. This centralized cooperative caching, which is converted into the form of a multiple-choice knapsack, is solved by using greedy algorithm for approximate optimal results. In [23], caching in the RUs assisted by broadcast TrS and TrA techniques is optimized under the constraints on storage and transmission rate by using a heuristic algorithm. This way can efficiently enhance the system throughput. Numerical results show that the proposed cooperative caching in the RUs gains the lowest latency, and thus it can be applied to delay-sensitive applications.

Interestingly, the work in [4] considers caching in the MBS to cooperate with caching in the RUs. Based on the mobility of VUs and the popularity of contents, an arbitrary content can be cached in both MBS and RUs, only MBS, or only RUs, so as to gain low transmission delay and service cost.

As a multi-objective multi-dimensional multi-choice knapsack problem, an ant colony optimization-based algorithm is used for near-optimal solution. Similarly, caching in MBS and RUs is also proposed in the studies [5], [6], [25] to optimize the latency, delay, and hit ratio. To address the challenges posed by high-dimensional state and hybrid continuous–discrete action spaces, a deep deterministic policy gradient-based approach is employed to solve the latency optimization problem [5]. Meanwhile, a combination of asynchronous federated learning (FL) and deep reinforcement learning (DRL) is utilized to further protect the privacy of VUs and enhance the hit ratio [6]. And in [25], to address the complexity arising from mobility and polularity prediction, data privacy, and dynamic cooperative caching, the authors simultaneously employed long short-term memory (LSTM), hierarchical FL, and adaptive gradient descent algorithm to enhance the caching performance in terms of hit ratio and delay.

Furthermore, the DC methods can be deployed in both RUs and VUs, or purely in VUs [21], [32]–[37]. In [21], the authors analyzed the features of contents (popularity and size), mobility, and traffic density to optimally cache in the RUs and VUs by applying cross entropy algorithm. The objective is to improve the hit ratio, reduce the delay, and remain low overhead. Meanwhile, the authors in [32] leveraged the social relationship among VUs to minimize the access latency through DRL and the authors in [33] utilized the clustering technique to maximize the hit ratio and reduce the delay by employing LSTM-based reinforcement learning. Similarly, the objective of the work [34] is to improve the hit ratio and reduce the delay while remaining low cost. A cooperation-based greedy algorithm is also proposed to solve the joint caching in the RUs and VUs in large-scale systems with low time complexity. When caching is performed exclusively in the VUs, incentive policy, privacy protection, and high accuracy are strictly required to enhance the caching performance. These issues can be addressed by Stackelberg game [35] and FL-based DQN [36], [37].

We can see that as shown in Table I, the mentioned DC methods are implemented in different configurations, including RUs only, RUs and MBS, RUs and VUs, or VUs only. They all leverage mobility, popularity and other assisted techniques to enhance the content delivery performance. However, the problem is that most of them do not exploit the available storage resources of VANETs to cache the contents in the VUs nor the benefits of PC methods. In addition, the number of RUs deployed is relatively small, without fully investigating the convergence, stability, and accuracy of solutions (except for [18]). As a result, the current DC methods fail to efficiently utilize the caching storage resources or improve the system performance.

C. PC Methods

In contrast to DC methods that rely on predefined caching placements in the MBSs, RUs, and VUs, PC methods adopt dynamic strategies in the VUs using probabilistic models [22], [27]–[31]. So, the PC methods allow the caching VUs to join or leave the system randomly to meet the real-time and dynamic conditions of VANETs. Particularly in [27], the authors proposed a distributed PC method by considering the demand and importance of VUs, popularity, and relative movement to determine caching probabilities. NS-3 based simulator is implemented to show that the proposed method can improve the cache hit ratio while ensuring low delay and high caching utilization. Similarly, the work in [28] further considers the privacy ratings of VUs to achieve not only low delay and high cache hit ratio, but also reduced cache redundancy, by conducting simulations in the OMNeT++ environment.

The PC method can be used for serving the common mobile users (MUs) as studied in [22]. To formulate the energy efficiency based PC optimization problem, this method employs Markov process to model the interactions between the VUs and the MUs. The popularity of contents is also considered to make the solution more efficient. The CoT technique from the MBS and VUs to the MUs is utilized to improve the system gain. Nonlinear fractional programming and Lyapunov optimization theory are used to solve the problem for optimal caching probability with high hit ratio and energy efficiency. In [31], the social relationship of VUs is taken into account besides the mobility of VUs and the popularity of contents. The RUs are responsible for collecting the movement information of VUs and updating the VUs on new contents by hidden Markov model. Opportunistic network environment simulator is deployed to demonstrate the benefit of the proposed PC method with social relationship in terms of hit ratio, average access delay, average hop counts, and average storage usage.

The joint solution of named data networking and probabilistic spatial content caching is studied in [30] to propose a communication scheme for content delivery in VANETs. Interestingly, a convolutional neural network (CNN) is applied to capture the complicated relationship between the mobility of VUs and the content requests. The CNN-based PC method is formulated to optimize the caching probabilities for achieving a better target of hit ratio and delay compared to non-CNN approach. Another promising extension of PC is the incorporation of both moving and parked VUs into caching decision policy as explored in [29]. This work pays attention to traffic density, road centrality, and popularity of contents for making decision of which probability to cache a content in the VUs. Simulation results show the effectiveness of the proposed solution with high hit ratio and low content retrieval delay.

Concerning the aforementioned PC methods [22], [27]–[31], we can see in Table I that most of them are deployed in simple systems by caching in the VUs only. The MBS and RUs are in charge of cooperatively transmitting the contents [22] or collecting the mobility of VUs [31]. Furthermore, the convergence, stability, and accuracy of the proposed algorithms and solutions have not been investigated, and thus the optimal results remain unconvincing. Although the social relationship of VUs and CNN approach is leveraged [30], [31], the lack of utilizing the powerful caching storage resource and stable wireless communications of the RUs in VANETs makes the conventional PC methods become less efficient.

III. System Model

In this paper, we consider a DPC model associated with the main notations for VASs in VANETs as shown in Fig. 1 and

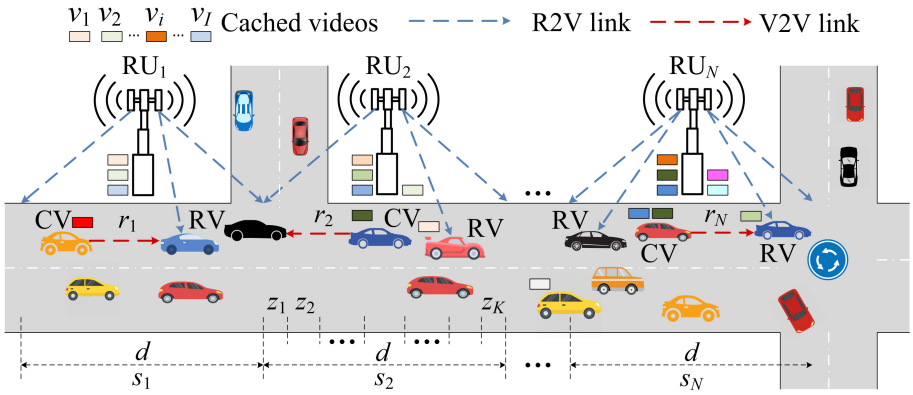

Fig. 1. DPC model for VASs in VANETs.

TABLE II. NOTATIONS

Symbols	Specifications
N	Number of RUs
K	Number of zones covered by each RU
I	Number of videos
S_i	Size of video i, $i = 1, 2, ..., I$
h_i	Caching radius of video i in DC
ρ_i	Caching probability of video i in PC
s_n	Average speed of VUs in road segment n covered by RU n, $n = 1, 2, ..., N$
r_n	Transmission range of vehicles covered by RU n
d	Distance of a road segment
λ	Density of vehicles
τ	Percentage of vehicles which are willing to act as CVs
α	Skewed access rate (popularity) exponent among videos
η	Path loss exponent
κ	Rician factor
P	Transmission power of CVs
N_0	Additive white Gaussian noise power

Table II. The model includes N RUs and I videos. In addition, a number of VUs are spatially distributed according to an independently thinned one-dimensional homogeneous Poisson point process (T1D-PPP) Φ with density λ [46], [47]. Each RU covers a road segment of d meters divided into K zones. The videos can be cached in both the RUs and VUs. The VUs, which enter the road segment n, $n = 1, 2, ..., N$, should drive at an average speed of s_n kilometers per hour (km/h). There are two types of VUs: caching VUs (CVs) which are willing to cache the videos and requesting VUs (RVs) which request the videos.

To facilitate the implementation of the proposed model, but without loss of generality, we assume that the RUs and VUs have limited storage resources available for caching; therefore, a constraint is introduced to ensure efficient resource utilization. Additionally, given the selfish nature of the VUs, an incentive mechanism is applied to motivate them to cache the popular videos [35]. In terms of channel state information (CSI), we further assume that the statistical knowledge of the CSI can be periodically estimated by the RUs [48], [49]. In this context, to improve the QoS of VASs in VANETs, a DPC optimization problem is formulated and solved for the optimal video caching placement in the RUs by DC and the optimal video caching probability deployed in the CVs by PC. The objective is to minimize the service time while utilizing the caching storage resources of RUs and CVs. The service time is defined as the average duration for the RVs to receive the requested videos from the RUs and the CVs while on the move. After deploying the DPC method, suppose an RV requests video i, $i = 1, 2, ..., I$, and enters a road segment covered by RU n, it is served by one of the following two situations:

1) If RU n does not cache video i, the RV receives video i from any CVs caching video i within a given transmission range r_n over V2V communications.
2) If RU n caches video i, it has the opportunity to serve the RV over R2V communications in cooperation with the CVs.

It is noted that if video i (i.e., of large size) cannot be completely transmitted after the RVs traverse all N RUs, its transmission will be continued by a subsequent set of RUs associated with another MBS. This scenario is beyond the scope of this paper and is left for future investigation.

A. Homogeneous Poisson Point Process for VUs

For the VUs, without loss of generality, we apply the T1D-PPP to model the spatial distribution of the CVs for transmitting and the RVs for receiving on the one-way roads [46], [47]. Let τ and ρ_i respectively be the percentage of VUs acting as CVs and the caching probability of video i, the distributions of CVs and RVs also follow the T1D-PPPs Φ_i^{CV} and Φ_i^{RV} of densities λ_i^{CV} and λ_i^{RV}, expressed as

$$\lambda_i^{\mathrm{CV}} = \tau\rho_i\lambda, \tag{1}$$

$$\lambda_i^{\mathrm{RV}} = (1 - \tau\rho_i)\lambda. \tag{2}$$

B. DC and PC

In DC, along the road consisting of N RUs, we define the caching placement of video i by a caching radius h_i, which is measured in hops and is inversely proportional to the caching density [50]. It means that video i is cached in the RUs separated by every h_i hops. If $h_i = 1$, video i is cached in all RUs. Meanwhile, in PC, we find the probability ρ_i to cache

video i in the CVs. While driving, the RVs are cooperatively served by both RUs and CVs using the DPC method, which considers the popularity pattern of video i, following a Zipf-like distribution [51], given by

$$p_i = \frac{i^{-\alpha}}{\sum_{i=1}^{I} i^{-\alpha}}, \tag{3}$$

where $\alpha \geq 0$ is the skewed access rate coefficient representing the popularity pattern of a set of videos, e.g., $\alpha = 0$ yields the same popularity for all videos.

C. V2V Communications

1) V2V Capacity: If video i is requested by an arbitrary RV, in the worst case, it takes the RV a driving time to go through $h_i - 1$ hops for reaching the RU where video i is cached. During the driving time, the RV is served by the CVs. In the T1D-PPP model, the probabilities that a typical CV caches and a typical RV requests video i within the range r_n in segment n are given respectively by [47]

$$p_{i,n}^{\mathrm{CV}} = 1 - e^{-2r_n\lambda_i^{\mathrm{CV}}}, \tag{4}$$

and

$$p_{i,n}^{\mathrm{RV}} = 1 - e^{-2r_n\lambda_i^{\mathrm{RV}}}. \tag{5}$$

The V2V channel between CVs and RVs is modelled as a Rician-based line-of-sight small-scale fading [52]–[54]. The channel capacity between CV a ($a \in \Phi_i^{\mathrm{CV}}$) and RV b ($b \in \Phi_i^{\mathrm{RV}}$) is therefore given by

$$C_{a,b} = W \log_2\left(1 + \frac{P|h|^2 d_{a,b}^{-\eta}}{N_0}\right), \tag{6}$$

where W is the system bandwidth, η is the path loss exponent, and $d_{a,b}$ is the distance from CV a to RV b. Also, h denotes the channel expressed as

$$h = \sqrt{\frac{\kappa}{1+\kappa}} e^{j\theta} + \sqrt{\frac{1}{1+\kappa}}\omega, \tag{7}$$

where κ is the Rician factor, θ is the random variable uniformly distributed in the range $[0, 2\pi]$, and ω is the complex Gaussian random variable with a zero-mean and unit-variance $\mathcal{CN}(0,1)$.

In this paper, we assume that within the given transmission range r_n, only one CV, which caches video i, is responsible for serving the associated RV over D2D communications in overlay mode [55], [56]. Based on (6) and in the worst case, the capacity for transmitting video i to an arbitrary RV over the maximum transmission range $d_{a,b} = r_n$, is given by

$$C_{i,n} = p_{i,n}^{\mathrm{CV}} p_{i,n}^{\mathrm{RV}} W \log_2\left(1 + \frac{P|h|^2 r_n^{-\eta}}{N_0}\right). \tag{8}$$

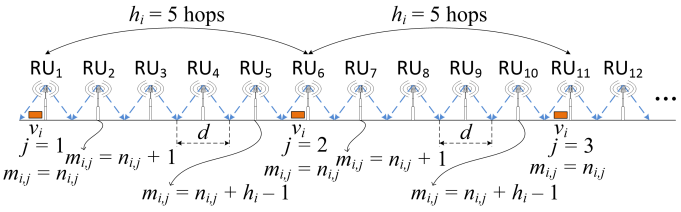

Fig. 2. An example of caching placement in DC with $h_i = 5$.

TABLE III. ZONES COVERED BY AN RU

Zones	1	2	3	4	5	6	7
d_k(m)	25	30	40	60	40	30	25
R_k(Mbps)	1	2	5.5	11	5.5	2	1

2) V2V Transmission Time: We can see that the number of RUs caching video i is given by

$$J_i = \left\lfloor \frac{N-1}{h_i} \right\rfloor + 1, \tag{9}$$

where the operator $\lfloor . \rfloor$ is used to return the greatest integer less than or equal to a given real number.

In other words, video i is cached in RU $n_{i,j} = h_i(j-1)+1$, $j = 1, 2, ..., J_i$. Once the RVs move out of the coverage of RU $n_{i,j}$, the time to receive video i from the CVs in segment $m_{i,j}$, $m_{i,j} = n_{i,j} + 1, n_{i,j} + 2, ..., n_{i,j} + h_i - 1$, is given by

$$t_{i,m_{i,j}} = \frac{d}{s_{m_{i,j}}}. \tag{10}$$

Fig. 2 explains (10) in case of $h_i = 5$. We assume that there is no transmission gap between two adjacent RUs. However, if the caching radius h_i is greater than 1 hop, there is a caching gap between the two adjacent RUs which cache video i.

D. R2V Communications

1) R2V Capacity: To derive the R2V capacity, we apply R2V transmission model given in [4]. In this model, the range d covered by an RU is divided into K zones. The parameters of zone k, $k = 1, 2, ..., K$, $K = 7$, are listed in Table III. The capacity per RV requesting video i in zone k served by RU n is given by

$$R_{i,n}^{(k)} = \begin{cases} 0, & \text{if } n \neq n_{i,j}, \\ \frac{(1-p_{i,n}^{\mathrm{CV}}) p_{i,k}^{\mathrm{RV}} R_k}{V_{i,k}}, & \text{if } n = n_{i,j}, \end{cases} \tag{11}$$

where $p_{i,k}^{\mathrm{RV}}$ and $V_{i,k}$ are respectively the probability that a typical RV in zone k requests video i from RU n and the number of RVs requesting video i within zone k, expressed as

$$p_{i,k}^{\mathrm{RV}} = 1 - e^{-d_k\lambda_i^{\mathrm{RV}}}, \tag{12}$$

and

$$V_{i,k} = d_k \lambda_i^{\mathrm{RV}}. \tag{13}$$

Algorithm 1 Computing t_i.

Input: t_i
$t_i = 0$
Output: t_i
1: **for** $j = 1 : J_i$ **do**
2: $n_{i,j} = h_i(j-1)+1$
3: **for** $k = 1 : K$ **do**
4: **if** $S_i > 0$ **then**
5: $S_i = S_i - t_{n_{i,j},k}(C_{i,n_{i,j}} + R^{(k)}_{i,n_{i,j}})$
6: $t_i = t_i + t_{n_{i,j},k}$
7: **else**
8: **break**
9: **end if**
10: **end for**
11: **for** $m_{i,j} = n_{i,j}+1 : n_{i,j}+h_i+1$ **do**
12: **if** $S_i > 0$ **and** $m_{i,j} \le N$ **then**
13: $S_i = S_i - t_{i,m_{i,j}} C_{i,m_{i,j}}$
14: $t_i = t_i + t_{i,m_{i,j}}$
15: **else**
16: **break**
17: **end if**
18: **end for**
19: **end for**

2) R2V Transmission Time: In addition, the transmission time of RU n to serve the RVs in zone k, i.e., the so-called average time to drive throughout zone k covered by RU n, is simply computed as

$$t_{n,k} = \frac{d_k}{s_n}. \tag{14}$$

E. Average Service Time

The overall average service time for I videos, which is the objective function to be minimized by finding h_i and ρ_i, is given by

$$\overline{T} = \sum_{i=1}^{I} p_i t_i, \tag{15}$$

where t_i is the average time to cooperatively receive video i from the CVs and the RUs presented in Alg. 1.

In Alg. 1, line 1 checks all the RUs, from the first to the last, which cache video i. Line 3 to line 10 are used for transmitting video i in different zones of the corresponding RU. If the transmission is incomplete within the RU's coverage, the CVs take over and continue transmitting video i as the RVs move beyond the RU (lines 11–18). It is noted that t_i is cumulatively added in line 6 and line 14 while the transmission of video i is still ongoing (i.e., $S_i > 0$).

IV. DPC Optimization Problem and GA Solution

A. DPC Optimization Problem

Based on the objective function (15) and by further taking the constraint on the caching storage resource of RUs and CVs into account, the DPC optimization problem is formulated as

$$\min_{h_i,\rho_i} \overline{T}, \tag{16a}$$

$$\text{s.t. } 1 \le h_i \le N-1, \forall i, \tag{16b}$$

$$S^{\mathrm{CR}} + S^{\mathrm{CV}} \le \delta N \textstyle\sum_{i=1}^{I} S_i, \tag{16c}$$

where (16c) is used to limit the total storage consumption for caching in both RUs and CVs with $0 < \delta < 1$, and S^{CR} and S^{CV} are the caching storage resources consumed by RUs and CVs, respectively computed as

$$S^{\mathrm{CR}} = \sum_{i=1}^{I} J_i S_i, \tag{17}$$

and

$$S^{\mathrm{CV}} = \tau \lambda d (N-1) \sum_{i=1}^{I} \rho_i S_i. \tag{18}$$

Algorithm 2 GA for DPC.

Input: System and GA parameters in Tables IV
$Gen = 1$: Generation count
Output: $T_F^*(h_i^*, \rho_i^*)$
1: Randomly generate
1) N_{P} sub-strings $\{X_1^{(z)}\}$, $z = 1, 2, ..., N_P$, each of I integer optimization variables $\{h_i\}$.
2) N_{P} sub-strings $\{X_2^{(z)}\}$, each of $I \times P_{\mathrm{R}}$ bits to characterize a set of I real optimization variables $\{\rho_i\}$, here P_{R} is the number of bits used to represent the precision of a real optimization variable which is valued at $2^{P_{\mathrm{R}}}$ quantization levels.
2: Compute N_{P} fitness values based on (16) to have $\{T_F^{(z)}\}$, i.e., $T_F^{(z)}\big(X_1^{(z)}, X_{2,\mathrm{R}}^{(z)}\big)$.
3: **while** TC does not hold **do**
4: Put $\{X^{(z)}\} = [\{X_1^{(z)}\}\{X_2^{(z)}\}]$ associated with $T_F^{(z)}$ into the mating pool for ranking.
5: Select $N_{\mathrm{PG}} = N_{\mathrm{P}} \times P_{\mathrm{G}}$ best individuals with lowest fitness values by using stochastic universal sampling operator [57] for breeding to obtain $\{X^{(t)}\} = [\{X_1^{(t)}\}\{X_2^{(t)}\}]$, $t = 1, 2, ..., N_{\mathrm{PG}}$.
6: Choose a pair of parents to create the offsprings by using single point crossover with probabilities P_{ci} and P_{cr} applied to $X_1^{(t)}$ and $X_2^{(t)}$, respectively.
7: Mutate the offsprings $\{X_1^t\}$ and $\{X_2^t\}$ respectively with probabilities P_{mi} and P_{mr} by using complement operation so that the positive genetic features probably lost in the previous steps can be recovered to obtain $\{X^{(t),*}\} = [\{X_1^{(t),*}\}\{X_2^{(t),*}\}]$.
8: Repeat step 2 to obtain $T_F^{(t),*}\big(X_1^{(t),*}, X_{2,\mathrm{R}}^{(t),*}\big)$.
9: Reinsert $\{X^{(t),*}\}$ and $T_F^{(t),*}$ into the present generation to obtain the new sets of $\{X^{(z)}\}$ and $T_F^{(z)}$.
10: $Gen = Gen + 1$
11: **end while**
12: Find the best fitness $T_F^*(h_i^*, \rho_i^*) \in \{T_F^{(z)}\big(X_1^{(z)}, X_{2,\mathrm{R}}^{(z)}\big)\}$.

B. GA Solution

GA has demonstrated strong capabilities in solving complex optimization problems across various domains [58], ranging

from communication systems [59]–[61] to VASs [62]–[65]. The key strengths of GA lie in its foundation on the evolutionary principles of natural selection and genetic variation. This enables GA to flexibly provide either exact or near-optimal global solutions, regardless of whether the search space is unimodal or multimodal. It means that GA excel at escaping local optima by simultaneously exploring multiple peaks in the search space [62], [63]. Thanks to its robustness, adaptability, and powerful global search capability, GA has become a widely adopted approach for solving real-world complex optimization problems.

In this paper, we apply GA [57] to solve (16). However, the problem is that GA supports only simple constraints in the form of lower and upper bounds, such as (16b), but not more complex ones like (16c). This problem is addressed by using the penalty method [62]. To this end, we convert (16) into an unconstrained optimization problem by reformulating (16c) as

$$\Delta S = \delta N \sum_{i=1}^{I} S_i - (S^{\mathrm{CR}} + S^{\mathrm{CV}}) \geq 0. \tag{19}$$

Then, we derive the penalty function as

$$F = \rho\big(\min\{0, \Delta S\}\big)^2, \tag{20}$$

where ρ is the constraint violation degree used to adjust the degree of punishment if the individuals in GA violate the constraints.

Finally, GA is capable of solving the following unconstrained DPC optimization problem

$$\min_{h_i, \rho_i} T_F = \overline{T} + F. \tag{21}$$

The detailed GA used to solve (21) is presented in Alg. 2. In Alg. 2, the two integer and real optimization variables cannot be represented by the same chromosome (string) within an individual, nor can they be handled using the same crossover and mutation schemes. Therefore, they are separately processed in two sub-strings $X_1^{(z)}$ and $X_2^{(z)}$ of individual z. It is noted that for $X_1^{(z)}$, a base-$(N-1)$ operation [57] is used to generate the integers from 0 to $(N-2)$, and then added by 1 to satisfy the constraint (16b). Furthermore, for $X_2^{(z)}$, it is converted into real optimization variables $X_{2,\mathrm{R}}^{(z)}$ by using the $\mathrm{b2r}$ operation [57]. GA is implemented by a sequence of main operators including reproduction/selection, crossover, and mutation, repeatedly until satisfying one of the two termination conditions (TC) given as follows:

- The average penalty value per individual ($\overline{F}$) derived from (20) is less than 10^{-3} in 10 consecutive generations.
- Generation count (Gen) is equal to a given number of generations ($N_{\mathrm{G}} = 100$).

It is evident that, as an adaptive heuristic search algorithm, the complexity of Alg. 2 is directly influenced by the size of search space (N_P) and the number of iterations (N_G). Meanwhile, the selected values of N_P and N_G can be large or small, relying on 1) the scale of system (N and I), 2) the computational complexity of objective function, constraints, and optimization variables in (16), and 3) the accuracy requirement for VASs. In addition, the complexity of Alg. 2 depends on the selection, crossover, and mutation operators. As a result, it is generally of the order of $\mathcal{O}(N_P N_G)$ [59], [63].

Commonly, it is more feasible for Alg. 2 to be executed by a single MBS, which then applies the DPC method across all RUs and CVs. However, in a larger-scale system with higher values of N and I, the main challenges in practically implementing the DPC method are twofold. First, it becomes difficult to collect the system information, such as the CSI, required to formulate the DPC optimization problem. Second, the complexity of Alg. 2 increases significantly. To address these challenges, the number of RUs managed by an MBS must be carefully determined based on the system scale and the processing capability of the MBS. It is further noted that if the requested videos are too large, as discussed in Section III, multiple MBSs can cooperate to continuously serve the RVs [25].

TABLE IV. SYSTEM AND GA PARAMETERS

Symbols	Specifications
N	25 RUs
K	7 zones [4]
I	10 videos
S_i	Uniformly random distributed from 10 to 200 (Mbps)
s_n	Uniformly random distributed from 30 to 60 (km/h)
r_n	Uniformly random distributed from 50 to 200 (m)
d	250m [4]
λ	0.025 vehicles/m
τ	0.2, 20% of VUs opt to act as CVs
δ	0.5, 50% of the total RU storage capacity is set as the upper bound for combined caching by RUs and VUs
α	1, skewed popularity exponent among videos [8], [19]
η	4, path loss exponent [10]
κ	4, Rician factor [52]
P	0.1W
N_0	10^{-9}W
ρ	10^{-2}, constraint violation degree properly determined based on the approach proposed in [63]
N_{P}	3000 individuals
N_{G}	100 generations
P_{G}	0.9, 90% of the individuals with the lowest fitness values are selected for breeding
P_{R}	5 bits, using 2^5 quantization levels to represent the real optimization variable ρ_i in the range [0, 1]
$\{P_{\mathrm{ci}}, P_{\mathrm{cr}}\}$	$\{0.6, 0.8\}$, set to sufficiently high values to ensure effective crossover and convergence
$\{P_{\mathrm{mi}}, P_{\mathrm{mr}}\}$	$\{10^{-12}, 10^{-10}\}$, set to sufficiently low values to ensure effective mutation and convergence

V. PERFORMANCE EVALUATION

A. Parameters Setting

The system and GA parameters are detailed in Table IV. To evaluate the performance of DPC method, we compare it to the other four schemes, including only caching in VUs (OCV) [22], only caching in RUs (OCR) [20], DPC with average per individual (DAV), and DPC with worst individual (DWO). In OCV and OCR, PC and DC are separately deployed in VUs and RUs and combined with the CoT technique, making them equivalent to the approaches in [22] and [20], respectively, as shown in Table I. Meanwhile in DAV and DWO, GA is applied to DPC optimization problem, but instead of finding the best individual as presented in Alg. 2, we find all the individuals

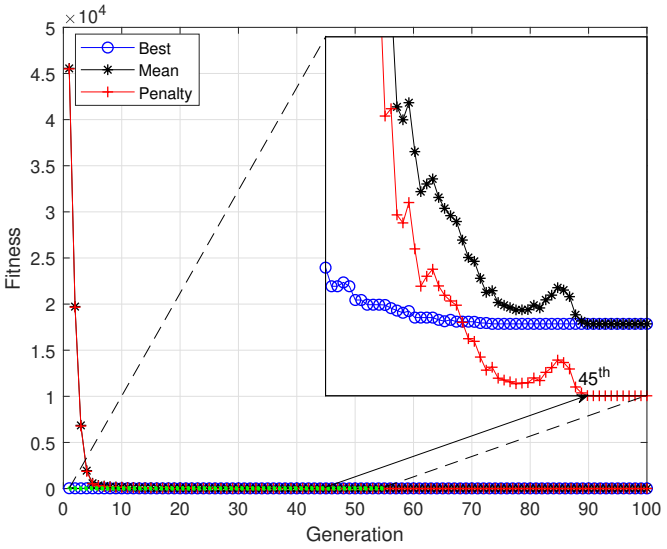

Fig. 3. GA convergence rate.

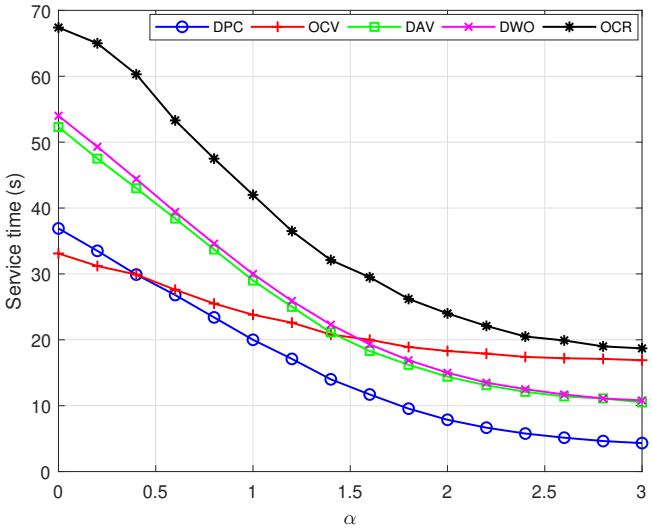

Fig. 5. Service time versus α.

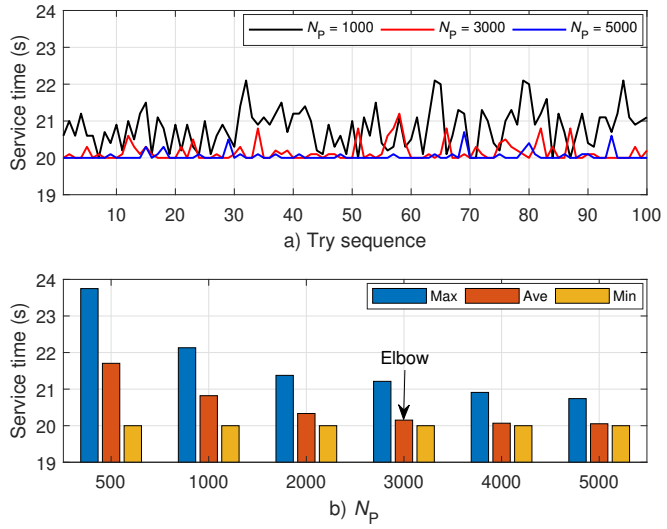

Fig. 4. GA stability and accuracy.

that satisfy the constraints (16b) and (16c). Then, we compute $\overline{T}$ in (15) w.r.t each individual. The overall average value of $\overline{T}$ per individual and the highest value of $\overline{T}$ in all individuals are obtained for DAV and DWO, respectively. Furthermore, we force the OCV, OCR, DAV, and DWO to cache as much storage as the DPC method does to ensure a fair comparison.

B. GA Convergence Evaluation

Fig. 3 plots the convergence rate of GA by computing the values of Best, Mean, and Penalty in each generation. Best represents the lowest fitness value associated with the best individual, while Mean and Penalty stand for the mean of fitness value and the mean of penalty value per individual. The results in Fig. 3 show that GA gets converged from the 45^{th} generation. In convergence situations, Mean equals Best leading to the fact that all individuals are characterized by the best genetic material. Simultaneously, Penalty decreases to zero to satisfy the constraints.

In Fig. 4, we further investigate the stability and accuracy of GA by carrying out it 100 tries w.r.t different population sizes (N_{P}). The stability of GA is realized in Fig. 4a) when increasing N_{P} from 1000, 3000, to 5000. The higher value of N_{P} we deploy, the higher stability GA achieves. The higher stability provides more possibilities of accurate results occurred, i.e., at $\overline{T} = 20$s, within 100 tries. The accuracy of GA is elaborated in Fig. 4b) by computing the maximum (Max), average (Ave), and minimum (Min) values of 100 tries for each N_{P}. We can see that the value of Min ($\overline{T} = 20$s) or the accurate result of GA always occurs within 100 tries even if the population size is very small ($N_{\text{P}} = 500$). Increasing N_{P} provides more accurate results and thus lower values of Ave and Max. However, we cannot deploy GA with a large population size due to high time and memory complexity. In this paper, the proper value of N_{P} is 3000 at the elbow point of Ave, which makes GA not only highly stable and accurate but also able to achieve reasonable complexity.

C. DPC Performance Evaluation

1) Service time: We evaluate the performance of DPC method and other schemes versus the skewed popularity exponent α as shown in Fig. 5. As α increases, a smaller number of videos become significantly more popular than the rest. Therefore, more storage resources are utilized for caching the most popular videos with higher caching probabilities in CVs and lower caching radiuses in RUs to obtain shorter service time. In this system, OCV plays a more important role in

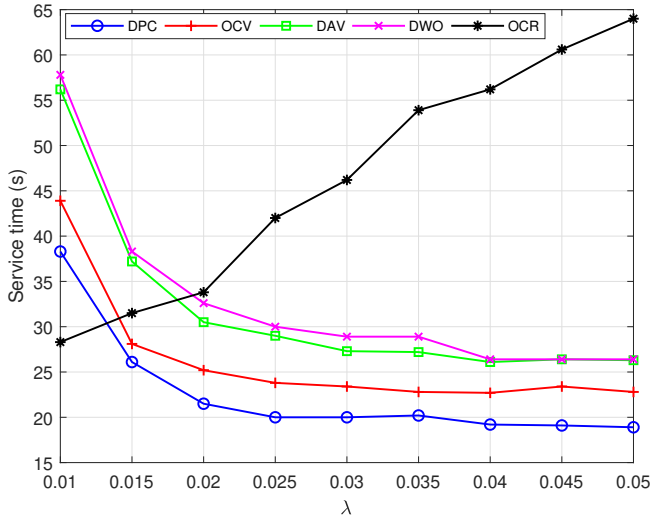

Fig. 6. Service time versus λ.

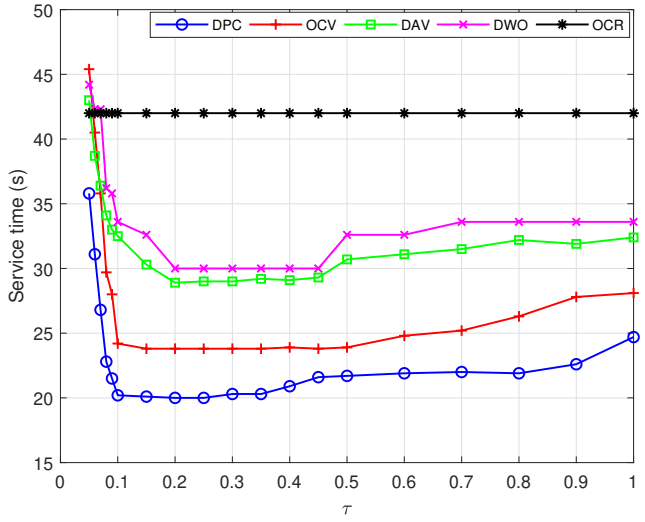

Fig. 7. Service time versus τ.

reducing the service time than OCR does. If α is too low ($\alpha \leq 0.4$), i.e., all videos have the same or similar popularity, it is difficult for DPC (a combination of OCV and OCR) to outperform the pure OCV because OCR is extremely bad. It is even impossible to make DPC equal to OCV due to the fact that in Fig. 9, GA cannot completely eliminate the role of such an extremely bad OCR from DPC by forcing the storage consumption of DPC-CR (S^{CR}) to be null and the storage of DPC-CV (S^{CV}) up to that of OCV. For the ease of deployment, the system designers prefer OCV to DPC if α is low. However, if α is high enough ($\alpha > 0.4$), the combination of OCV and OCR works well to make the DPC more efficient and thus it is much better than the others schemes. Owning the advantages of DPC, DAV and DWO provide lower service time compared to OCV and OCR.

In Fig. 6, the performance is investigated versus the density of VUs λ. For DPC, OCV, DAV, and DWO, they partially or completely rely on PC, and thus the service time decreases w.r.t the increase of λ. This is because increasing λ enables the system to cache the videos in the CVs more successfully for serving the RVs faster. The service time is reduced to a certain saturated value due to the limits of (4) and (5). Obviously, if λ is low ($\lambda \leq 0.0125$), the PC-based DPC, OCV, DAV, and DWO are not good and even worse than the OCR — an alternate solution for low density of VUs. However, for OCR, the service time gets higher w.r.t the increase of λ because of the division of channel capacity according to the number of RVs computed by (11). When λ is higher, it is important to conclude that DPC always surpasses the other schemes.

Fig. 7 illustrates the service time performance versus τ, i.e., the percentage of VUs which are willing to act as the CVs. We can easily reveal that the pure DC-based OCR keeps unchanged due to the simultaneous decrease in the numerator and denominator of (11) as τ increases. Meanwhile, the other PC-based DPC, OCV, DAV, and DWO show that there is an

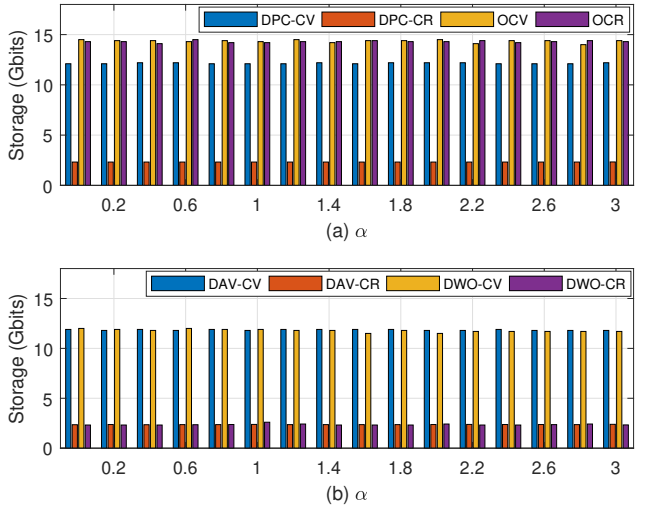

Fig. 8. Storage consumption versus α.

optimal range of τ for achieving the minimum service time. The optimal range of τ occurs when (4) and (5) are optimally balanced to maximize (8) or minimize the service time. This interesting finding provides a useful method to assign the optimal percentage of VUs acting as the CVs to serve the RVs with the fastest service time. In comparison, the proposed DPC gains the best performance in terms of service time if τ is high enough ($\tau \geq 0.05$). If τ is low ($\tau < 0.05$), DPC certainly approaches OCR, while the other OCV, DAV, and DWO become worse than OCR.

2) Storage consumption: For storage consumption evaluation, the total caching storage consumed by DPC method, which includes caching in CVs (DPC-CV) and caching in RUs

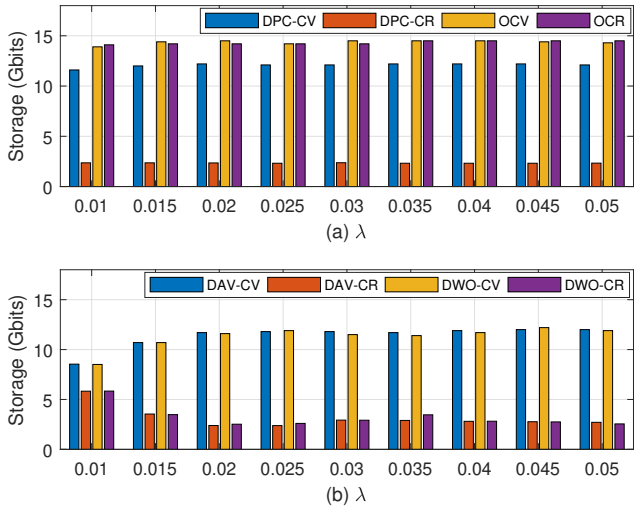

Fig. 9. Storage consumption versus λ.

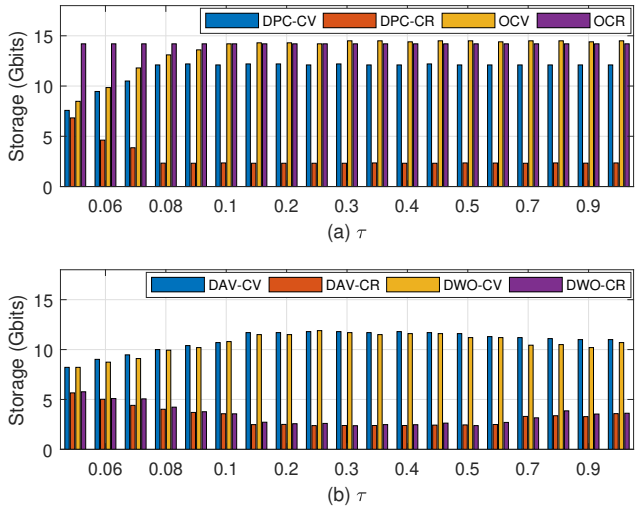

Fig. 10. Storage consumption versus τ.

(DCP-CR), is equivalently utilized by OCV, OCR, DAV, and DWO to ensure a fair comparison. It is noted that because DAV and DWO schemes are based on DPC method, the total caching storage consumption of DAV (or DWO) similarly includes DAV-CV and DAV-CR (or DWO-CV and DWO-CR). Fig. 8 plots the caching storage consumption versus α. We can see that in Fig. 8(a), the storage resources consumed by DPC (DPC-CV and DPC-CR), OCV, and OCR are equivalent. This also happens to DAV and DWO in the same manner of DPC as shown in Fig. 8(b).

Similarly, Fig. 9 plots the caching storage consumption versus λ. It can be seen that the storage resources consumed by all DPC, OCV, OCR, DAV, and DWO are still equivalent to ensure the fair comparison. Importantly, in Fig. 9(a), DPC optimally allocates a consistent percentage of storage consumption between DPC-CV and DPC-CR regardless of increasing λ for the fastest service time. Meanwhile, in Fig. 9(b), the storage resources consumed by DAV and DWO naturally change without any optimal storage allocation. DAV and DWO can exploit more available storage resources of CVs when λ becomes higher to provide faster service time by increasing the storage consumption of DAV-CV and DWO-CV but decreasing that of DAV-CR and DWO-CR.

Finally, we investigate the caching storage consumption versus τ as illustrated in Fig. 10. The results in Fig. 10(a) show that the optimal storage resources of DPC allocated to caching in CVs and in RUs are realized more explicitly. If τ is low ($\tau \leq 0.07$), the role of OCV is less important, thus stimulating the role of OCR. As a result, the difference in the storage consumption between DPC-CV and DPC-CR is not as significant as that shown in Fig. 9(a). DAV and DWO schemes in Fig. 10(b) are shown in a similar manner with DPC method in Fig. 10(a). It is clear that when τ is sufficiently high ($\tau > 0.07$), the behavior of DPC, OCV, OCR, DAV, and DWO in Fig. 10 becomes similar to those in Fig. 9.

VI. Conclusion

We have proposed the DPC model for VASs in VANETs. The DPC model exploits the advantages of DC and PC to cache the videos in the stationary RUs and the dynamic VUs, respectively. The combination of DC and PC enables the cooperative video transmission from the RUs and the CVs to serve the RVs efficiently. Furthermore, many features of RUs (quantity), VUs (density and speed), videos (popularity and size), CoT, and GA are studied to formulate the DPC optimization problem for finding the optimal results of caching radius and caching probability. The total caching storage resources of RUs and CVs are considered as a joint constraint in the DPC optimization problem so that the separated storages consumed by DC and PC are properly balanced depending on the situation of VANETs. GA is modified to deal with the complexity of integer and real optimization variables of the DPC problem. Consequently, simulation results are shown to demonstrate the convergence, stability and accuracy of GA and the benefits of DPC method in terms of minimum service time while utilizing the storage resource. The key findings are further discussed to provide deeper insights into the design of edge caching techniques for VASs in VANETs.

References

[1] X. Chen, Y. Deng, H. Ding, G. Qu, H. Zhang, and P. Li, "Vehicle as a service (VaaS): Leverage vehicles to build service networks and capabilities for smart cities," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 3, pp. 2048–2081, 3rd Quarter 2024.

[2] P. Laclau, S. Bonnet, B. Ducourthial, X. Li, and T. Lin, "Enhancing automotive user experience with dynamic service orchestration for software defined vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 26, no. 1, pp. 824–834, Jan. 2025.

[3] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel prediction based scheduling for data dissemination in VANETs," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1409–1412, Jan. 2017.

[4] J. Chen, H. Wu, P. Yang, F. Lyu, and X. Shen, "Cooperative edge caching with location-based and popular contents for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 291–10 305, Sep. 2020.

[5] H. Tian, X. Xu, L. Qi, X. Zhang, W. Dou, and S. Yu, "CoPace: Edge computation offloading and caching for self-driving with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13 281–13 293, Dec. 2021.

[6] W. Yang and Z. Liu, "Efficient vehicular edge computing: A novel approach with asynchronous federated and deep reinforcement learning for content caching in VEC," *IEEE Access*, vol. 12, pp. 13 196–13 212, Jan. 2024.

[7] S. A. Elsayed, S. Abdelhamid, and H. S. Hassanein, "Predictive proactive caching in VANETs for social networking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5298–5313, May 2022.

[8] T. C. Lam, N.-S. Vo, M.-P. Bui, C. D. T. Thai, H. Jung, and V.-C. Phan, "Service time-aware caching, power allocation, and 3D trajectory optimised multimedia content delivery in UAV-assisted IoT networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 4, pp. 6419–6432, Apr. 2025.

[9] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, and J. Shi, "Intelligent content caching strategy in autonomous driving toward 6G," *IEEE Trans. Intelligent Transportation Syst.*, vol. 23, no. 7, pp. 9786–9796, July 2022.

[10] Z. Lin, Y. Fang, P. Chen, F. Chen, and G. Zhang, "Modeling and analysis of edge caching for 6G mmWave vehicular networks," *IEEE Trans. Intelligent Transportation Syst.*, vol. 24, no. 7, pp. 7422–7434, July 2023.

[11] X. Zhou, M. Bilal, R. Dou, J. J. P. C. Rodrigues, Q. Zhao, and J. Dai, "Edge computation offloading with content caching in 6G-enabled IoV," *IEEE Trans. Intelligent Transportation Syst.*, vol. 25, no. 3, pp. 2733–2747, Mar. 2024.

[12] Y. Wang, H. Dai, X. Han, P. Wang, Y. Zhang, and C. Xu, "Cost-driven data caching in edge-based content delivery networks," *IEEE Trans. Mobile Computing*, vol. 22, no. 3, pp. 1384–1400, Mar. 2023.

[13] T. C. Lam, N.-S. Vo, V. V. Lam, T. Hoang, M.-P. Bui, and T. Q. Duong, "Multi-rate selection and power allocation assisted probabilistic edge caching for cooperative video transmission in dense D2D networks," *Alexandria Engineering Journal*, vol. 126, pp. 555–564, July 2025.

[14] S. Zhang, T. Cai, D. Wu, D. Schupke, N. Ansari, and C. Cavdar, "IoRT data collection with LEO satellite-assisted and cache-enabled UAV: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5872–5884, Apr. 2024.

[15] M.-H. T. Nguyen, T. T. Bui, L. D. Nguyen, E. Garcia-Palacios, H.-J. Zepernick, and H. Shin, "Real-time optimized clustering and caching for 6G satellite-UAV-terrestrial networks," *IEEE Trans. Intelligent Transportation Syst.*, vol. 25, no. 3, pp. 3009–3019, Mar. 2024.

[16] Y. Zhao, C. Liu, X. Hu, J. He, M. Peng, and D. W. K. Ng, "Joint content caching, service placement, and task offloading in UAV-enabled mobile edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 1, pp. 51–63, Jan. 2025.

[17] H. Wang, J. Han, K. Xue, J. Yang, J. Li, and Q. Sun, "CAAF: An NDN-based cache-aware adaptive forwarding strategy for reliable content delivery in VANETs," *IEEE Trans. Mobile Computing*, pp. 1–15, Feb. 2025, Early access.

[18] T. Cao, Z. Zhang, X. Wang, H. Xiao, and C. Xu, "PTCC: A privacy-preserving and trajectory clustering-based approach for cooperative caching optimization in vehicular networks," *IEEE Trans. Sustainable Computing*, vol. 9, no. 4, pp. 615–630, July-Aug. 2024.

[19] Y.-T. Wang, T.-Y. Lin, S.-I. Sou, L.-A. Chen, M.-H. Tsai, and Y.-R. Chen, "Markov clustering-based content placement in roadside-unit caching with deadline constraint," *IEEE Trans. Intell. Transport. Syst.*, vol. 25, no. 9, pp. 11 881–11 892, Sep. 2024.

[20] Z. Jin, T. Song, W. Jiang, and J. Hu, "A centralized edge cooperative caching strategy for VANETs," in *Proc. IEEE Wireless Commun. and Networking Conf.*, Dubai, United Arab Emirates, Apr. 2024, pp. 1–6.

[21] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, and Y. Jia, "An edge caching scheme to distribute content in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5346–5356, June 2018.

[22] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3100–3112, Apr. 2019.

[23] S. Berri, J. Zhang, B. Bensaou, and H. Labiod, "Joint content-prefetching, transmission scheduling, and rate adaptation in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4348–4358, Apr. 2022.

[24] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug. 2018.

[25] B. Feng, C. Feng, D. Feng, Y. Wu, and X.-G. Xia, "Proactive content caching scheme in urban vehicular networks," *IEEE Trans. Commun.*, vol. 71, no. 7, pp. 4165–4180, July 2023.

[26] Y. AlNagar, R. H. Gohary, S. Hosny, and A. A. El-Sherif, "Mobility-aware edge caching for minimizing latency in vehicular networks," *IEEE Open J. Veh. Technol.*, vol. 3, pp. 68–84, Feb. 2022.

[27] G. Deng, L. Wang, F. Li, and R. Li, "Distributed probabilistic caching strategy in VANETs through named data networking," in *Proc. IEEE Int. Conf. Computer Commun. Workshops*, San Francisco, CA, Apr. 2016, pp. 1–6.

[28] W. Zhao, Y. Qin, D. Gao, C. H. Foh, and H.-C. Chao, "An efficient cache strategy in information centric networking vehicle-to-vehicle scenario," *IEEE Access*, vol. 5, pp. 12 657–12 667, June 2017.

[29] S. A. Elsayed, S. Abdelhamid, and H. S. Hassanein, "Probabilistic cooperative caching in VANETs for social networking," in *Proc. IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[30] G. Manzo, E. Kalogeiton, A. D. Maio, T. Braun†, M. R. Palattella, and I. Turcanu, "DeepNDN: Opportunistic data replication and caching in support of vehicular named data," in *Proc. IEEE Int. Symp. A World of Wireless, Mobile and Multimedia Netw.*, Cork, Ireland, Aug. 2020, pp. 234–243.

[31] L. Yao, Y. Wang, X. Wang, and G. WU, "Cooperative caching in vehicular content centric network based on social attributes and mobility," *IEEE Trans. Mobile Computing*, vol. 20, no. 2, pp. 391–402, Feb. 2021.

[32] H. Wu, Y. Fan, J. Jin, H. Ma, and L. Xing, "Social-aware decentralized cooperative caching for internet of vehicles," *IEEE Internet of Things J.*, vol. 10, no. 16, pp. 14 834–14 845, Aug. 2023.

[33] R. Wang, Z. Kan, Y. Cui, D. Wu, and Y. Zhen, "Cooperative caching strategy with content request prediction in Internet of vehicles," *IEEE Internet of Things J.*, vol. 8, no. 11, pp. 8964–8975, June 2021.

[34] Z. Xue, Y. Liu, G. Han, F. Ayaz, Z. Sheng, and Y. Wang, "Two-layer distributed content caching for infotainment applications in VANETs," *IEEE Internet of Things J.*, vol. 9, no. 3, pp. 1696–1711, Feb. 2022.

[35] C. Wang, C. Chen, Q. Pei, N. Lv, and H. Song, "Popularity incentive caching for vehicular named data networking," *IEEE Trans. Intelligent Transportation Syst.*, vol. 23, no. 4, pp. 3640–3653, Apr. 2022.

[36] C. Li, Y. Zhang, and Y. Luo, "A federated learning-based edge caching approach for mobile edge computing-enabled intelligent connected vehicles," *IEEE Trans. Intelligent Transportation Syst.*, vol. 24, no. 3, pp. 3360–3369, Mar. 2023.

[37] H. Wu, B. Wang, H. Ma, X. Zhang, and L. Xing, "Multiagent federated deep-reinforcement-learning-based collaborative caching strategy for vehicular edge networks," *IEEE Internet of Things J.*, vol. 11, no. 14, pp. 25 198–25 212, July 2024.

[38] J. Huang, D. Fang, Y. Qian, and R. Q. Hu, "Recent advances and challenges in security and privacy for V2X communications," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 244–266, June 2020.

[39] B. Bandyopadhyay, P. Kuila, and M. C. Govil, "Popularity-conscious service caching and offloading in digital twin and NOMA-aided con-

nected autonomous vehicular systems," *IEEE Trans. Netw. and Service Management*, vol. 21, no. 6, pp. 6451–6464, Dec. 2024.

[40] N. R.R., G. Ghatak, V. A. Bohara, and A. Srivastava, "Performance analysis of cache-enabled handover management for vehicular networks," *IEEE Trans. Netw. Science and Engineering*, vol. 11, no. 1, pp. 1151–1164, Jan.-Feb. 2024.

[41] Y. Guo, Q. Yang, F. R. Yu, and V. C. M. Leung, "Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5445–5459, June 2018.

[42] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Computational Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.

[43] H. Yan, X. Xu, M. Bilal, X. Xia, W. Dou, and H. Wang, "Customer centric service caching for intelligent cyber–physical transportation systems with cloud–edge computing leveraging digital twins," *IEEE Trans. Consumer Electronics*, vol. 70, no. 1, pp. 1787–1797, Feb. 2024.

[44] J. Zheng, T. H. Luan, G. Li, Z. Yin, Y. Wu, and M. Dong, "ACDV: Adaptive content delivery for vehicular digital twin networks," *IEEE Trans. Veh. Technol.*, pp. 1–16, Mar. 2025, Early access.

[45] B. Mao, Y. Liu, J. Liu, and N. Kato, "AI-assisted edge caching for metaverse of connected and automated vehicles: Proposal, challenges, and future perspectives," *IEEE Veh. Technol. Mag.*, vol. 18, no. 4, pp. 66–74, Dec. 2023.

[46] J. P. Jeyaraj, M. Haenggi, A. H. Sakr, and H. Lu, "The transdimensional poisson process for vehicular network analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8023–8038, Dec. 2021.

[47] S. Zhao, Y. Zhang, Y. Zhu, Z. Zhao, and Y. Liu, "Optimal probabilistic collaborative caching in UAV-assisted vehicular networks," in *Proc. IEEE Int. Conf. Intell. Transportation Syst.*, Macau, China, Oct. 2022, pp. 1–6.

[48] M. Asefi, J. W. Mark, and X. S. Shen, "A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over VANETs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1817–1827, May 2012.

[49] B. Feng, C. Feng, G. Min, and T. Q. S. Quek, "Two-timescale adaptive live video streaming transmission mechanism for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 4, pp. 6823–6828, Apr. 2025.

[50] N.-S. Vo, T. Q. Duong, L. Shu, X. Du, H.-J. Zepernick, and W. Cheng, "Cross-layer design for video replication strategy over multihop wireless networks," in *Proc. IEEE Inter. Commun. Conf.*, Kyoto, Japan, Jun. 2011, pp. 1–6.

[51] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proc. IEEE Int. Conf. Computer Commun.*, New York, NY, Mar. 1999, pp. 126–134.

[52] S. Zhu, T. S. Ghazaany, S. M. R. Jones, R. A. Abd-Alhameed, J. M. Noras, and T. V. Buren, "Probability distribution of Rician K-factor in urban, suburban and rural areas using real-world captured data," *IEEE Trans. Antennas and Propagation*, vol. 62, no. 7, pp. 3835–3839, July 2014.

[53] B. Hu, L. Fang, X. Cheng, and L. Yang, "Vehicle-to-vehicle distributed storage in vehicular networks," in *Proc. IEEE Inter. Commun. Conf.*, Kansas City, MO, May 2018, pp. 1–6.

[54] ——, "In-vehicle caching (IV-Cache) via dynamic distributed storage relay (D^2SR) in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 843–855, Jan. 2019.

[55] Q.-N. Tran, N.-S. Vo, Q.-A. Nguyen, M.-P. Bui, T.-M. Phan, V.-V. Lam, and A. Masaracchia, "D2D multi-hop multi-path communications in B5G networks: A survey on models, techniques, and applications," *EAI Endorsed Trans. Industrial Netw. and Intell. Syst.*, vol. 7, no. 25, pp. 1–12, Jan. 2021.

[56] Y. Pei and Y.-C. Liang, "Resource allocation for device-to-device communication overlaying two-way cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3611–3621, Jul. 2013.

[57] A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca, *"Genetic Algorithm TOOLBOX For Using with Matlab"*. Ver 1.2 Users Guide, University of Sheffield, 1994.

[58] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, p. 8091–8126, 2021.

[59] X. Qi, S. Khattak, A. Zaib, and I. Khan, "Energy efficient resource allocation for 5G heterogeneous networks using genetic algorithm," *IEEE Access*, vol. 9, pp. 160 510–160 520, Nov. 2021.

[60] H. Li *et al.*, "Energy-efficient task offloading of edge-aided maritime UAV systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1116–1126, Jan. 2023.

[61] M. I. Mushtaq *et al.*, "Framework for optimized resource allocation in multi-user, multi-service, multi-device aerial networks," *IEEE Access*, vol. 12, pp. 54 866–54 878, Apr. 2024.

[62] T. Fang and L. P. Chau, "GOP-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks," *IEEE Trans. Image Processing*, vol. 15, no. 6, pp. 1323–1330, Jun. 2006.

[63] N.-S. Vo, T. Q. Duong, H. D. Tuan, and A. Kortun, "Optimal video streaming in dense 5G networks with D2D communications," *IEEE Access*, vol. 6, pp. 209–223, Oct. 2017.

[64] H. Zhang, X. Cao, J. K. L. Ho, and T. W. S. Chow, "Object-level video advertising: An optimization framework," *IEEE Trans. Industrial Informatics*, vol. 13, no. 2, pp. 520–531, Apr. 2017.

[65] G. Lai, F. F. Leymarie, and W. Latham, "On mixed-initiative content creation for video games," *IEEE Trans. Games*, vol. 14, no. 4, pp. 543–557, Dec. 2022.