

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC DUY TÂN

NGUYỄN QUỲNH ANH

**TỔNG HỢP CÁC CÔNG BỐ LIÊN QUAN
LUẬN ÁN TIẾN SĨ KỸ THUẬT**

**CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ CHUYÊN NGÀNH: 9.48.01.01**

**LƯU TRỮ VÀ PHÂN PHỐI TÀI NGUYÊN TỐI ƯU CHO
DỊCH VỤ TRUYỀN VIDEO TRONG UAV-VANET**

ĐÀ NẴNG, 2026

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC DUY TÂN

NGUYỄN QUỲNH ANH

**TỔNG HỢP CÁC CÔNG BỐ LIÊN QUAN
LUẬN ÁN TIẾN SĨ KỸ THUẬT**

**CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ CHUYÊN NGÀNH: 9.48.01.01**

**LƯU TRỮ VÀ PHÂN PHỐI TÀI NGUYÊN TỐI ƯU CHO
DỊCH VỤ TRUYỀN VIDEO TRONG UAV-VANET**

ĐÀ NẴNG, 2026

DANH MỤC CÁC BÀI BÁO LIÊN QUAN ĐẾN LUẬN ÁN TIẾN SĨ

[1] Nguyen-Son Vo, **Quynh-Anh Nguyen**, Thanh-Minh Phan, Thanh-Hieu Nguyen, and ThanhDung Tran, "Caching Placement Minimised Service Time for Video Streaming over Roadside Units in VANETs," *Journal of Aviation Science & Technology*, vol. 1, no. 2, pp. 1-6, Nov. 2022.

[2] **Quynh-Anh Nguyen**, Nguyen-Son Vo, Thuong C. Lam, Tan Do-Duy, Haejoon Jung, and Trung Q. Duong, "Joint Deterministic and Probabilistic Edge Caching Minimized Service Time for Cooperative Video Transmission in VANETs," *IEEE Trans. Netw. Science and Engineering*, vol. 13, pp. 1932-1943, Sep. 2025.

[3] **Quynh-Anh Nguyen**, Nguyen-Son Vo, Van-Ca Phan, Dac-Binh Ha, Minh-Phung Bui, and Long D. Nguyen, "Joint Caching and Hovering Minimized Service Time for Cooperative Video Streaming in UAV-assisted VANETs", *REV Journal on Electronics and Communications*, vol. 16, no. 1, pp. 24-35, Jan.-Mar. 2026.

Caching Placement Minimised Service Time for Video Streaming over Roadside Units in VANETs

Nguyen-Son Vo^{1*}, Quynh-Anh Nguyen², Thanh-Minh Phan³, Thanh-Hieu Nguyen²,
and Thanh-Dung Tran⁴

¹Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, Vietnam

²Ho Chi Minh City University of Transport, Ho Chi Minh City, Vietnam

³Vietnam Aviation Academy, Ho Chi Minh City, Vietnam

⁴Absolute Software Company, Ho Chi Minh City, Vietnam

*Corresponding Author / E-mail: vonguyenson@duytan.edu.vn

Manuscript received: July, 2022 / Revised: October 01, 2022 / Accepted: November 15, 2022

ABSTRACT

Video streaming services (VSSs) are considerably requested by on-the-road users (RUs) in vehicular ad-hoc networks (VANETs). Gaining high quality of experience (QoE) of VSSs is more challenging due to dynamic characteristics of the RUs and complicated parameters of the videos. In this paper, we exploit the available storage and processing capacity of roadside units (RSUs) to improve the QoE of VSSs in VANETs. To do so, a caching placement (CAP) optimisation problem is formulated for optimal caching radius (measured in hops) of each video. The optimal solution is found in accordance with the patterns of not only the arrival and service at the RSUs but also the access rate of the videos, so as to minimise the service time for video streaming over RSUs in VANETs. Simulation results are presented to demonstrate the advantages of the CAP solution compared to other conventional schemes.

KEYWORDS: Caching networks, quality of experience, vehicular, video streaming applications and services

1. Introduction

Recently, we have witnessed an extreme demand for advanced applications and services (AASs) by enormous Internet of things devices, machines, vehicles, and augmented/virtual reality and mobile users. It is anticipated that there will be nearly 300 million of AASs used by 12.3 billion of mobile subscribers in the upcoming years [1]. Among various AASs, video streaming services (VSSs) are occupying up to 79% of the mobile data traffic [2]. This traffic together with complicated characteristics of video contents and sensitive criteria of quality of experience (QoE) pose a set of challenges to VSSs providers.

In the same time, the computing and communication technologies have been rapidly developed to provide on-the-road users (RUs) with various vehicular services and applications for safety and entertainment such as image-aided navigation, automated driving, intelligent control, and VSSs [3]. It is more challenging for the VSSs providers in the context of vehicular ad-hoc networks (VANETs) in which most of the features of the system are highly dynamic, especially the RUs. To deal with these challenges, caching is considered as a promising technique that not only improves the QoE of VSSs in VANETs but also relaxes workloads at the backhaul links and

importantly saves the investment cost of system architecture upgrades [4], [5].

Caching techniques for tasks and contents can be deployed with/without other techniques depending on the objectives of the AASs [3], [6]–[11]. In [3], the authors studied the task offloading and service caching in vehicular edge computing (VEC). The edge servers installed at the roadside units (RSUs) cache the executed services that can be reused by the upcoming tasks to maximise the offloading efficiency satisfying a given service deadline. Caching-assisted VEC has been also studied in [6], [7] in which the authors considered an energy consumption constraint while minimising the response time of the AASs.

Interestingly, a collaborative data caching and computation offloading optimisation solution for multi-service VEC was proposed to maximise the storage utilisation with low latency and energy consumption [8]. In [9], a more detailed solution that exploits the computing, caching and communication resources of the smart vehicles in proximity and the RSUs to minimise the total network delay under long-term energy constraint. For content caching only, the authors in [11] proposed a cooperative caching for two types of content requests including location-based and popular contents to minimise the overall transmission delay and service cost. To

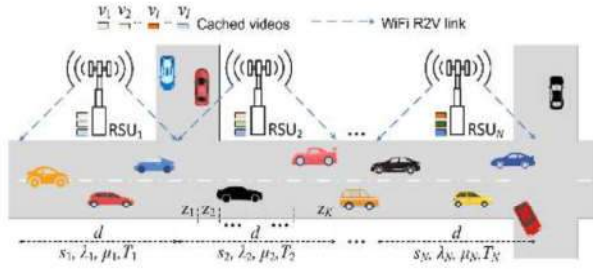


Figure 1. CAP model for VSSs in VANETs

Table 1. Notations

Symbols Specifications

N	Number of RSUs
K	Number of zones in the coverage range of an RSU
I	Number of videos
d	Distance of a road segment covered by an RSU
d_k	Distance of zone k in a road segment, $k = 1, 2, \dots, K$
R_k	Transmission rate of zone k
S_i	Size of video i measured in bits, $i = 1, 2, \dots, I$
p_i	Access rate (popularity) of video i
λ_n	Average arrival rate of vehicles at RSU n , $n = 1, 2, \dots, N$
μ_n	Service rate served by RSU n
s_n	Average speed of vehicles in the range covered by RSU n
T_n	Average waiting time in queue to be served by RSU n
h_i	Caching radius of video i measured in hops
α	Skewed access rate (popularity) exponent coefficient among different videos

further improve the offloading efficiency, caching technique is combined with broadcast scheduling and rate adaptation based on RSU-to-vehicle (R2V) communications [10].

However, the caching techniques in the aforementioned studies, which are relatively complicated, have not considered both the patterns of the arrival and service times at the RSUs and the access rate of the videos to improve the system performance. In this paper, we propose a simple but efficient caching placement (CAP) optimisation solution deployed in the RSUs to minimise the service time - one of the key performance metrics to improve the QoE of VSSs in VANETs. Particularly, a CAP optimisation problem is formulated and then solved for the optimal caching radius measured in hops, i.e., the hop distance between two adjacent RSUs in which the requested video is cached. The main idea of CAP solution is how to find the optimal caching radius of each video depending on the two important patterns, so as to minimise the service time. To do so, a CAP optimisation problem is formulated for finding the optimal caching radius of each video under a given caching storage capacity. This problem is solved by using genetic algorithms (GAs). The results are presented to demonstrate the

feasibility of GAs and the benefits of CAP solution compared to other conventional schemes.

The rest of this paper is organised as follows. We introduce the system model and describe how it works in Section 2. In Section 3, the system is formulated to derive the objective function of the CAP optimisation problem. Based on the system formulations, Section 4 presents the CAP optimisation problem and solution by using GAs. Simulation results are analysed and discussed in Section 5. Finally, we conclude the paper in Section 6.

2. System Model

In this paper, we consider a CAP model and related notations for VSSs in VANETs as shown in Figure 1 and Table 1, respectively. The model includes N RSUs and I videos. Each RSU covers a road segment of d meters divided into K zones. The vehicles, which enter the road segment n , $n = 1, 2, \dots, N$, should drive at an average speed of s_n kilometers per second and be served by the RSU n . An arbitrary vehicle can request the video i , $i = 1, 2, \dots, I$, in any RSUs at time t . The vehicle is served, i.e., queueing and receiving the requested video, immediately if it is covered by the RSU caching the requested video, otherwise it has to drive for several road segments before being served.

In this context, to improve the QoE of VSSs, a CAP optimisation problem is formulated and solved for the optimal caching placement of each video in the RSUs, so as to minimise the service time. The service time is defined as driving time and serving time. Here, the serving time, which stands for queueing and receiving time, is alternately used for waiting and transmitting time. In case the video i cannot be completely transmitted after the vehicles pass all N RSUs, it is continuously transmitted by the corresponding MBS. This context is out of the scope of the paper and we will be dedicated to studying in our future work.

3. System Formulations

3.1. M/M/1 queue based delay

At each RSU in VANETs, we assume that the arrival of vehicles and the job service times follow M/M/1 queue model determined by a Poisson process and an exponential distribution, respectively. The arrival and service times are considered as independent and identically distributed random variables for the ease of modelling. In this model, we define 1) λ_n , $n = 1, 2, \dots, N$, be the average arrival rate of vehicles covered by the RSU n ; 2) μ_n be the service rate served by the RSU n . It is easy to derive the average queuing delay, i.e., waiting time, of each vehicle given by

$$T_n = \frac{1}{\mu_n - \lambda_n}, \quad (1)$$

where $\mu_n > \lambda_n$ to avoid the overload situation at the RSU n .

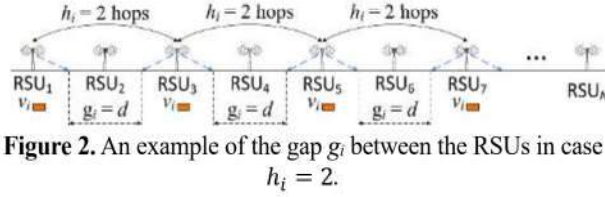


Figure 2. An example of the gap g_i between the RSUs in case $h_i = 2$.

Table 2. Zones at an RSU

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

3.2. Video caching placement

In general, the caching placement for a particular video can be found based on caching density, caching probability, or determinate caching location. In this paper, along the road of N RSUs, we define the caching placement of the video i represented by h_i which is a caching radius measured in hops and inversely proportional to the caching density r_i [12]. It means that the video i is cached at the RSUs separated by every h_i hops. If $h_i = 1$, the video i is cached in all RSUs.

3.3. Driving Time

If the video i is requested by a vehicle at the RSU n , it takes this vehicle a driving time $D_{i,n}$ to go for reaching the RSU where the video i is cached, which is given by

$$D_{i,n} = d \sum_{j=1}^{J_i} \sum_{n=n_{i,j}+1}^{n_{i,j}+h_i-1} \sum_{m=n}^{n_{i,j}+h_i-1} \frac{1}{s_m}, \quad (2)$$

where $J_i = \left\lceil \frac{N-1}{h_i} \right\rceil + 1$ is the number of RSUs which caches

the video i . In other words, the video i is cached at the RSU $n_{i,j} = h_i(j-1) + 1, j = 1, 2, \dots, J_i$. The eq. (2) is clearly explained Figure 2 for $h_i = 2$. We assume that the waiting time to be served at an arbitrary RSU is much less than the driving time throughout the RSU. We also assume that there is no transmission gap between two adjacent RSUs. However, if the caching radius h_i is greater than 1 hop, there is a gap between the two adjacent RSUs which cache the video i . Here, d is the standard gap coefficient associated with $h_i = 2$ hops.

So far, the overall average driving time is given by

Algorithm 1 Computing t_i

Input: $S_i, R_{k,n_{i,j}}$

$T_{X_i} = 0$

$T_{W_i} = 0$

Output: t_i

1: **for** $j = 1 : J_i$ **do**

2: $T_{W_i} = T_{W_i} + T_{n_{i,j}}$

3: **for** $k = 1 : K$ **do**

4: **if** $S_i > 0$ **then**

5: $S_i = S_i - t_{k,n_{i,j}} R_{k,n_{i,j}}$

6: $T_{X_i} = T_{X_i} + t_{k,n_{i,j}}$

7: **else**

8: $t_i = \min \left\{ T_{X_i} + T_{W_i}, \frac{d}{t_{k,n_{i,j}}} \right\}$

9: **break**

10: **end if**

11: **end for**

12: **end for**

$$\bar{D} = \sum_{i=1}^I p_i \sum_{n=1}^N P_{i,n} D_{i,n}, \quad (3)$$

where $P_{i,n}$ is the probability that the video i is requested by at least one vehicle arriving at the RSU n within a given investigated time t , computed as

$$P_{i,n} = 1 - \exp(-p_i \lambda_n t), \quad (4)$$

and p_i is the access rate (popularity) of the video i following Zipf-like distribution [13]

$$p_i = \frac{i^{-\alpha}}{\sum_{i=1}^I i^{-\alpha}}, \quad (5)$$

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 access rate for all videos.

3.4. Serving Time

The serving time t_i is defined as the waiting time T_{W_i} and transmission time T_{X_i} . To derive the transmission time at the RSUs, we apply R2V transmission model given in [14]. In this model, the range d covered by an RSU is divided into K zones. The information of the zone $k, k = 1, 2, \dots, K, K = 7$, is listed in Table 2. The transmission rate per vehicle in the zone k served by the RSU n is given by

$$R_{k,n} = \frac{R_k}{V_{k,n}}, \quad (6)$$

$$V_{k,n} = \frac{d_k V_n}{\sum_{k=1}^K d_k} = \frac{d_k V_n}{d}, \quad (7)$$

Table 3. Parameters setting

Symbols	Specifications
N	20 RSUs
K	7 zones [14]
I	10 videos
$\{S_i\}$	$\{30, 50, 20, 60, 10, 70, 80, 100, 90, 40\}$ (Mbps)
λ_n	Uniformly random distributed from 0.5 to 4 (vehicles/s)
μ_n	Uniformly random distributed from 1 to 5 (vehicles/s)
s_n	Uniformly random distributed from 50 to 100 (km/h)
d	250 (m)
α	1
δ	0.5

where V_n is the average number of vehicles covered by the RSU n given by

$$V_n = \lambda_n T_n. \quad (8)$$

In addition, the transmission time of the RSU n to serve the vehicles in the zone k , i.e., the so-called time to drive throughout the zone k of the RSU n , is computed as

$$t_{k,n} = \frac{d_k}{s_n}. \quad (9)$$

And the serving time t_i of the video i in the road segment of N RSUs is given in the Algorithm 1. So, the average serving time for all the videos is expressed as

$$\bar{T} = \sum_{i=1}^I p_i t_i. \quad (10)$$

Finally, the overall average service time, which is the objective function to be minimised by finding h_i , is given by

$$S_T = \bar{D} + \bar{T}. \quad (11)$$

4. CAP problem and GAs solutions

4.1. CAP Problem

Based on the objective function (11) and by further taking the constraint of storage capacity for caching into account, the CAP optimization problem is formulated as

$$\min_{h_i} S_T \quad (12a)$$

$$\text{s.t. } 1 \leq h_i \leq N-1, \forall i, \quad (12b)$$

$$\sum_{i=1}^I J_i S_i \leq \delta N \sum_{i=1}^I S_i. \quad (12c)$$

where (12c) is used to limit the caching storage consumption and $0 < \delta < 1$ (if $\delta = 1$, we obtain $h_i = 1; \forall i$).

4.2. GAs Solution

In this paper, we apply GAs [15] to solve (12). However, GAs only support the simple constraints like (12b), but not the complicated ones like (12c). To overcome this problem, we convert (12) to an unconstrained optimisation problem by using penalty method [16]. In this way, we rewrite (12c) as below

$$\Delta S = \delta N \sum_{i=1}^I S_i - \sum_{n=1}^N J_i S_i \geq 0. \quad (13)$$

We then derive the penalty function as

$$F = \lambda (\min\{0, \Delta S\})^2, \quad (14)$$

where λ is the constraint violation degree used to adjust the degree of punishment if the individuals in GAs violate the constraint.

Finally, the GAs are capable of solving the following unconstrained CAP problem

$$\min_{h_i} S_F = S_T + F. \quad (15)$$

The detailed GAs used to solve (15) are presented in [17]. However, because the GAs in [17] are implemented with real variables, while h_i in (15) is integer. So, in the initial generation, all individuals are randomly created by using a base $(N-1)$ operator. This way yields every individual in the form of an array of I elements, each element is an integer h_i in the range from 1 to $N-1$ (12b). The GAs are implemented by a sequence of main operators including reproduction/selection, crossover, and mutation, repeatedly until satisfying a given set of convergence criteria.

5. Performance evaluation

5.1. Parameters Setting

The system parameters are listed in Table 3. The observation time, which ensures the system stable enough, is selected to be $t = \max\{\frac{d}{s_n}, n = 1, 2, \dots, N\}$. For GAs implementation, we set the number of individuals in the population, the generation gap, the crossover probability, mutation probability, and the number of generations, respectively at 2500, 0.9, 0.8, 10^{-6} , and 50. Furthermore, λ is set at 0.1. The method to select λ is presented in [18].

5.2. GAs convergence evaluation

Figure 3 investigates the convergence rate of GAs within 50 generations. We can see that after several first generations being unstable, the GAs get started to converge from the 15th generation and completely converge from the 25th generation. In convergence situation, the penalty function F in (14) and (15) equals to zero to satisfy the constraint (12c). Furthermore, the best and the mean fitness values become closer and finally the

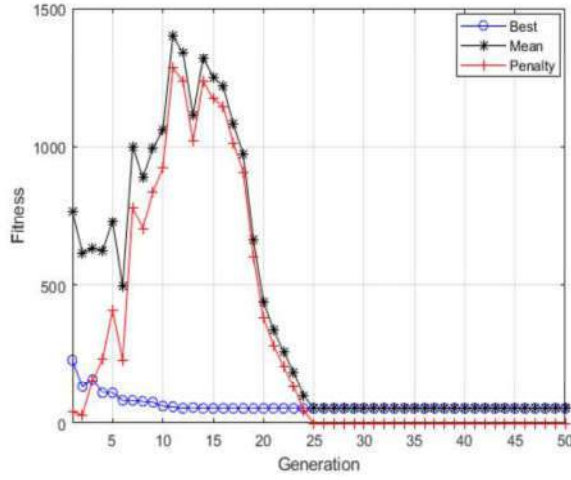
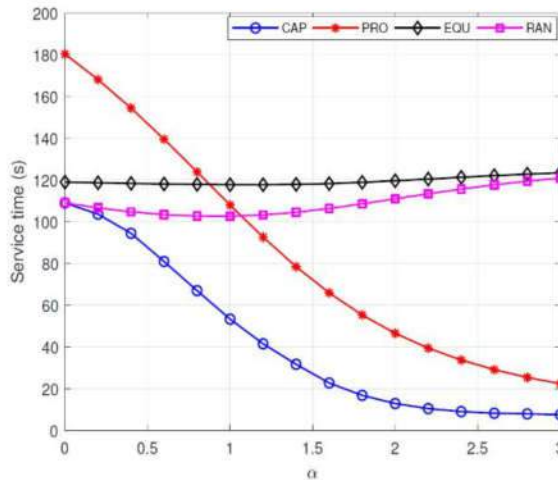


Figure 3. Convergence rate of GAs

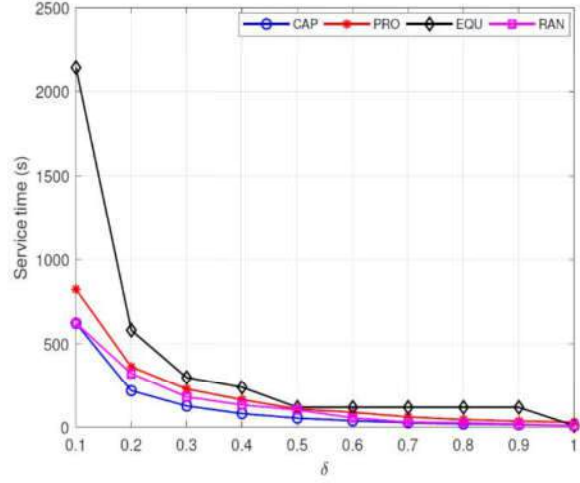
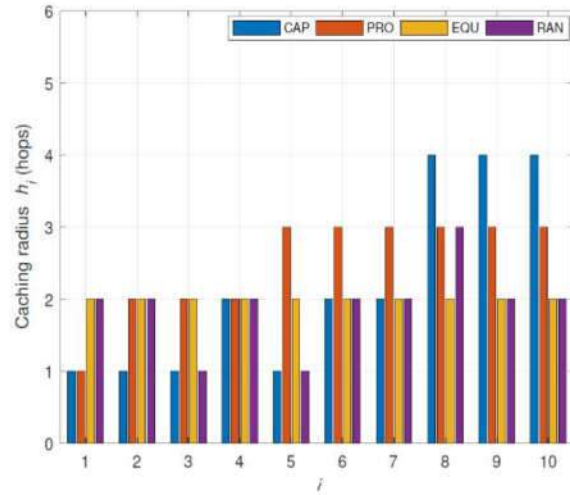
Figure 4. Service time versus α .

same following the evolutionary principles of GAs. The convergence result demonstrates the feasibility of GAs to solve the CAP optimisation problem.

5.3. CAP Performance Evaluation

To evaluate the performance of CAP, we compare it to 1) proportional (PRO), equal (EQU), and random (RAN) caching schemes, in which all h_i are found satisfying (12b) and (12c). In PRO, we set $h_i = \frac{n}{\sqrt{r_i}}$, here $r_i \propto p_i$ and Ω is selected to have proper values of h_i [12]. The EQU results in h_i to be the same for all videos. Meanwhile in RAN, h_i is randomly generated such that $\sum_{i=1}^I h_i$ is minimised. The detailed performance evaluation is presented in the sequel.

In Figure 4, we evaluate the service time of all schemes versus α . The results show that thanks to utilising both the patterns of the arrival and service at the RSUs and the access rate of the videos, CAP always outperforms the other PRO, EQU, and RAN schemes. In comparison with EQU and RAN, by exploiting the access rate pattern of the videos, PRO is better only if α is high enough, i.e., $\alpha > 1$. PRO and CAP pay more attention on the system patterns, which enable to significantly

Figure 5. Service time versus δ Figure 6. Caching radius h_i versus i .

decrease the service time. EQU and RAN are worst due to regardless of any system patterns. It is noted that RAN is better than EQU because among all the random individuals satisfying the constraint (12c), the one with $\sum_{i=1}^I h_i$ minimised is selected.

Figure 5 shows the service time versus the storage capacity constraint coefficient δ . It is certain that the less the storage capacity is used ($\delta \sim 0$), the longer the service time is attained. And it is also clear that if the storage capacity is fully used ($\delta = 1$), all the videos are cached in each RSU ($h_i = 1; \forall i$) leading to the results that the service time is shortest for all schemes, without optimisation solution. In comparison, the proposed CAP provides shorter service time than the PRO, EQU, and RAN do.

In consideration of the optimal results h_i shown in Figure 6, the optimal results exactly capture the characteristics of CAP, PRO, EQU, and RAN schemes that are optimal, proportional, equal, and random, respectively. Importantly, we can see that the distribution of h_i of PRO is similar to that of CAP if α and δ are high enough. In this case, PRO takes a high priority over CAP because the h_i of PRO is directly and quickly computed. Thus, alternately using CAP and PRO versus different scenarios of α and δ can be a proper solution for video streaming over RSUs with caching in VANETs.

6. Conclusion

We have proposed the CAP solution for VSSs in VANETs. The CAP exploits the available storage and processing capacity of the RSUs to improve the QoE of VSSs. In particular, the CAP optimisation problem is formulated and then being solved for optimal caching placement, i.e., caching radius of each video. The objective is to minimise the overall average service time represented by driving time and serving time. The optimal caching radius is found for minimum service time in accordance with both the arrival and service patterns of RSUs and the access rate pattern of videos. The results show that the proposed CAP solution outperforms the other conventional schemes in terms of service time under the same resource consumption constraint.

References

1. Cisco, "Cisco Annual Internet Report," in *2018–2023 White Paper*, Mar. 2020. [Online]. Available: <https://www.cisco.com..>
2. —, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update," in *2017–2022 White Paper*, Feb. 2019. [Online]. Available: <https://www.cisco.com>.
3. Q. Shen, B.-J. Hu, and E. Xia, "Dependency-aware task offloading and service caching in vehicular edge computing," *IEEE Trans. Veh. Technol.*, pp. 1–16, Aug. 2022, Early access.
4. T. Q. Duong, K. J. Kim, Z. Kaleem, M.-P. Bui, and N.-S. Vo, "UAV caching in 6G networks: A survey on models, techniques, and applications," *Physical Commun.*, vol. 51, pp. 1–19, Apr. 2022.
5. N.-T. Dinh, "An efficient traffic-aware caching mechanism for information-centric wireless sensor networks," *EAI Endorsed Trans. Industrial Netw. and Intell. Syst.*, vol. 9, no. 30, pp. 1–8, Apr. 2022.
6. C. Tang, C. Zhu, X. Wei, Q. Li, and J. J. P. C. Rodrigues, "Task caching in vehicular edge computing," in *Proc. IEEE Int. Conf. Computer Commun. Workshops*, Vancouver, BC, Canada, May 2021, pp. 1–6.
7. C. Tang, C. Zhu, H. Wu, Q. Li, and J. J. P. C. Rodrigues, "Toward response time minimization considering energy consumption in caching assisted vehicular edge computing," *IEEE Internet of Things J.*, vol. 9, no. 7, pp. 5051–5064, Apr. 2022.
8. H. Feng, S. Guo, L. Yang, and Y. Yang, "Collaborative data caching and computation offloading for multi-service mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9408–9422, Sep. 2021.
9. Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 4, pp. 2212–2225, Apr. 2021.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. N.-S. Vo, D.-B. Ha, B. Canberk, and J. Zhang, "Green two-tiered wireless multimedia sensor systems: An energy, bandwidth, and quality optimization framework," *IET Commun.*, vol. 10, no. 18, pp. 2543–2550, Dec. 2016.
18. 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.

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.

Index Terms—Caching networks, deterministic caching, probabilistic caching, vehicular ad-hoc networks (VANETs), 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], [4], [5], [6], [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], [10], [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, drones, and satellite networks [8], [12], [13], [14], [15], [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], [5], [6], [20], [21], [22], and broadcast transmission

Received 8 March 2025; revised 17 June 2025; accepted 31 August 2025. Date of publication 3 September 2025; date of current version 2 December 2025. This work was supported by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.04-2023.25. Recommended for development by Prof. Yuanwei Liu. (Corresponding author: Nguyen-Son Vo.)

Quynh-Anh Nguyen is with Vietnam Aviation Academy, Ho Chi Minh City 70000, Vietnam (e-mail: anhng@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 the 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 the Kyung Hee University, Yongin 02447, South Korea (e-mail: haejoonjung@khu.ac.kr).

Trung 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, BT7 1NN Belfast, U.K. (e-mail: tduong@mun.ca).

Digital Object Identifier 10.1109/TNSE.2025.3605822

TABLE I
 FEATURES OF DC AND PC CACHING METHODS

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

scheduling (TrS) and transmission rate adaptation (TrA) [23]) to address the highly dynamic environments 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], [5], [6], [19], [20], [23], [24], [25], [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], [28], [29], [30], [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], [39], [40], [41], [42], [43], [44], [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. 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], [43], [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], [5], [6], [18], [19], [20], [21], [23], [24], [25], [26], [32], [33], [34], [35], [36], [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 popularity 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], [33], [34], [35], [36], [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], [28], [29], [30], [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], [28], [29], [30], [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.

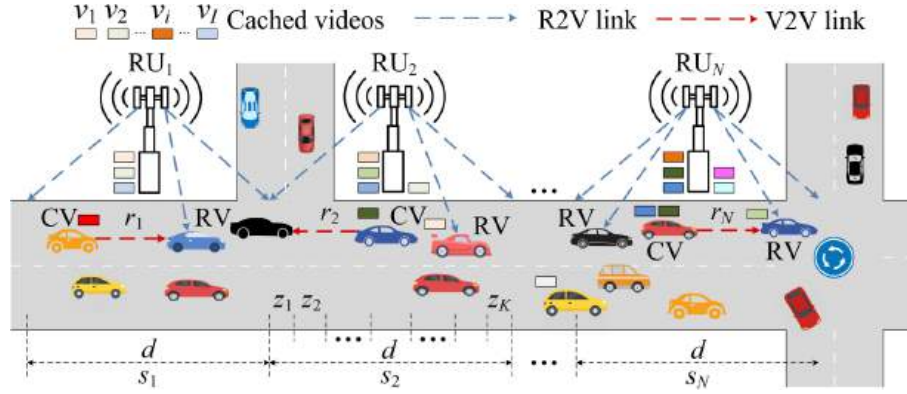


Fig. 1. DPC model for VASs in VANETs.

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 , $i = 1, 2, \dots, I$
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 , $n = 1, 2, \dots, N$
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

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 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 , $n = 1, 2, \dots, N$, 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 , $i = 1, 2, \dots, I$, 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

$$\lambda_i^{\text{CV}} = \tau \rho_i \lambda, \quad (1)$$

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

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

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

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]

$$p_{i,n}^{\text{CV}} = 1 - e^{-2r_n \lambda_i^{\text{CV}}}, \quad (4)$$

and

$$p_{i,n}^{\text{RV}} = 1 - e^{-2r_n \lambda_i^{\text{RV}}}. \quad (5)$$

The V2V channel between CVs and RVs is modelled as a Rician-based line-of-sight small-scale fading [52], [53], [54]. The channel capacity between CV a ($a \in \Phi_i^{\text{CV}}$) and RV b ($b \in \Phi_i^{\text{RV}}$) is therefore given by

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

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

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

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

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

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

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

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

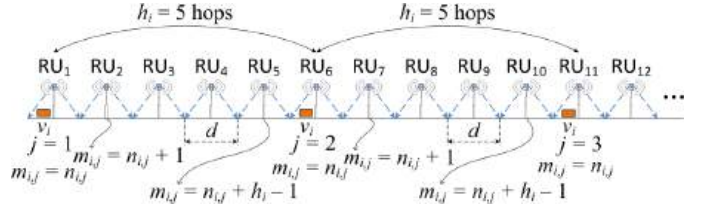


Fig. 2. An example of caching placement in DC with $h_i = 5$.

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

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

$$t_{i,m_{i,j}} = \frac{d}{s_{m_{i,j}}}. \quad (10)$$

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 , $k = 1, 2, \dots, K$, $K = 7$, are listed in Table III. The capacity per RV requesting video i in zone k served by RU n is given by

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

where $p_{i,n}^{\text{RV}}$ 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

$$p_{i,n}^{\text{RV}} = 1 - e^{-d_k \lambda_i^{\text{RV}}}, \quad (12)$$

and

$$V_{i,k} = d_k \lambda_i^{\text{RV}}. \quad (13)$$

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

$$t_{n,k} = \frac{d_k}{s_n}. \quad (14)$$

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:        $S_i = S_i - t_{n_{i,j},k}(C_{i,n_{i,j}} + R_{i,n_{i,j}}^{(k)})$ 
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  $m_{i,j} \leq N$  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

```

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

$$\bar{T} = \sum_{i=1}^I p_i t_i, \quad (15)$$

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

In Algorithm 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

$$\min_{h_i, \rho_i} \bar{T}, \quad (16a)$$

$$\text{s.t. } 1 \leq h_i \leq N-1, \forall i, \quad (16b)$$

$$S^{\text{CR}} + S^{\text{CV}} \leq \delta N \sum_{i=1}^I S_i, \quad (16c)$$

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  $\{X_1^{(z)}\}$ ,  $z = 1, 2, \dots, N_P$ , each of  $I$  integer optimization variables  $\{h_i\}$ .
   2)  $N_P$  sub-strings  $\{X_2^{(z)}\}$ , each of  $I \times P_R$  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  $2^{P_R}$  quantization levels.
2: Compute  $N_P$  fitness values based on (16) to have  $\{T_F^{(z)}\}$ , i.e.,  $T_F^{(z)}(X_1^{(z)}, X_2^{(z)})$ .
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  $N_{PG} = N_P \times P_G$  best individuals with lowest fitness values by using stochastic universal sampling operator [57] for breeding to obtain  $\{X^{(t)}\} = [\{X_1^{(t)}\}\{X_2^{(t)}\}]$ ,  $t = 1, 2, \dots, N_{PG}$ .
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  $T_F^{(t),*}(X_1^{(t),*}, X_2^{(t),*})$ .
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  $T_F^*(h_i^*, \rho_i^*) \in \{T_F^{(z)}(X_1^{(z)}, X_2^{(z)})\}$ .

```

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

$$S^{\text{CR}} = \sum_{i=1}^I J_i S_i, \quad (17)$$

and

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

B. GA Solution

GA has demonstrated strong capabilities in solving complex optimization problems across various domains [58], ranging from communication systems [59], [60], [61] to VASs [62], [63], [64], [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

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

Then, we derive the penalty function as

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

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

$$\min_{h_i, \rho_i} T_F = \bar{T} + F. \quad (21)$$

The detailed GA used to solve (21) is presented in Algorithm 2. In Algorithm 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 $X_{2,R}^{(z)}$ by using the 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 ($\bar{F}$) derived from (20) is less than 10^{-3} in 10 consecutive generations.
- Generation count (Gen) is equal to a given number of generations ($N_G = 100$).

It is evident that, as an adaptive heuristic search algorithm, the complexity of Algorithm 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 Algorithm 2 depends

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]
$\{P_{ci}, P_{cr}\}$	{0.6, 0.8}, set to sufficiently high values to ensure effective crossover and convergence
$\{P_{mi}, P_{mr}\}$	{ 10^{-12} , 10^{-10} }, set to sufficiently low values to ensure effective mutation and convergence

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 Algorithm 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 Algorithm 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].

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 Algorithm 2, we find all the individuals that satisfy the constraints (16b) and (16c). Then, we compute $\bar{T}$ in (15) w.r.t each individual. The overall average value of $\bar{T}$ per individual and the highest value of $\bar{T}$ in all individuals

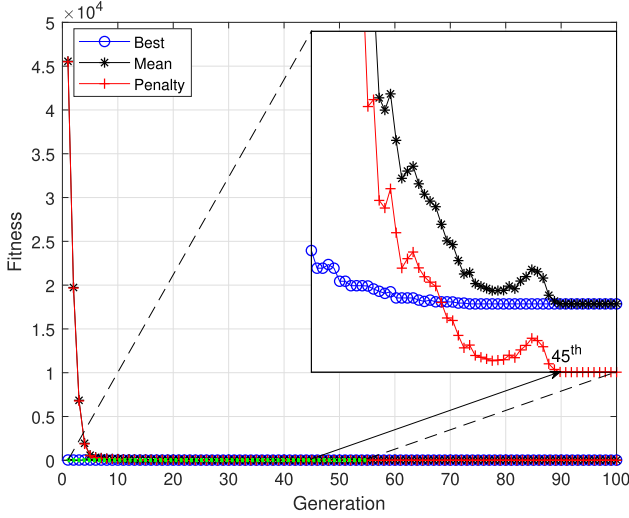


Fig. 3. GA convergence rate.

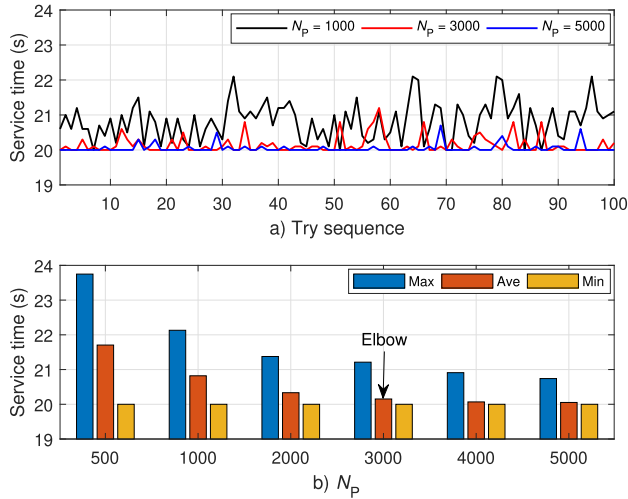


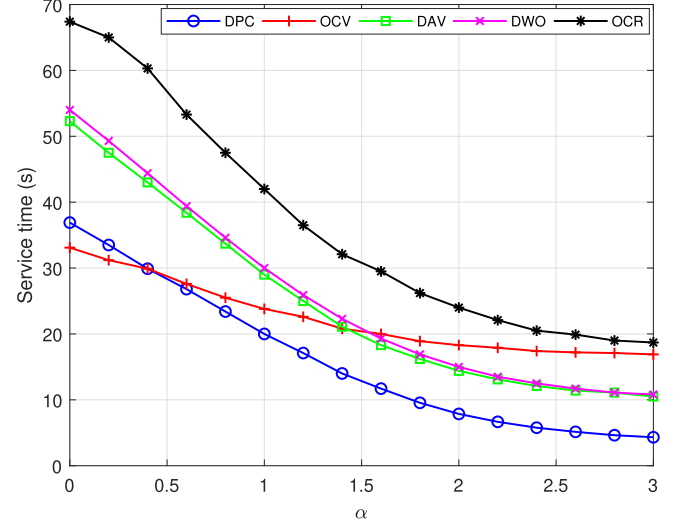
Fig. 4. GA stability and accuracy.

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 45th 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.

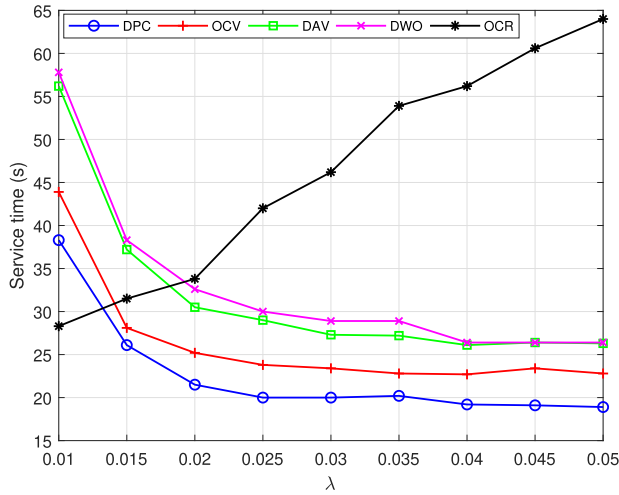
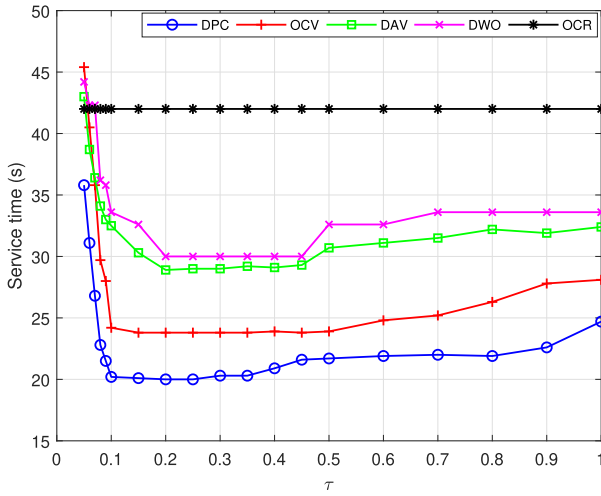
In Fig. 4, we further investigate the stability and accuracy of GA by carrying out it 100 tries w.r.t different population sizes (N_P). The stability of GA is realized in Fig. 4(a) when increasing N_P from 1000, 3000, to 5000. The higher value of N_P we deploy,

Fig. 5. Service time versus α .

the higher stability GA achieves. The higher stability provides more possibilities of accurate results occurred, i.e., at $\bar{T} = 20$ s, within 100 tries. The accuracy of GA is elaborated in Fig. 4(b) 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 ($\bar{T} = 20$ s) or the accurate result of GA always occurs within 100 tries even if the population size is very small ($N_P = 500$). 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.

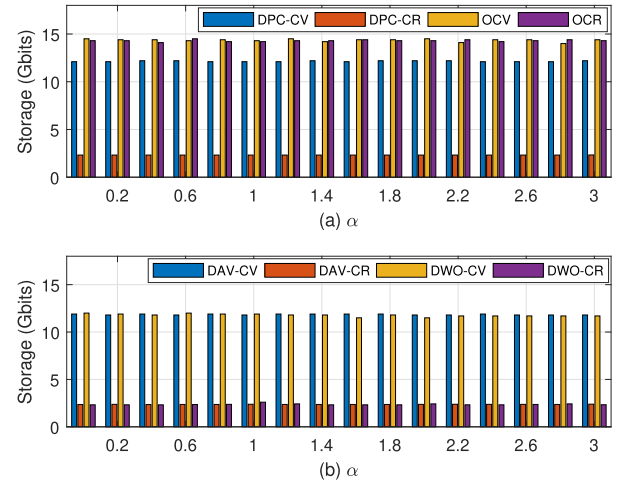
C. DPC Performance Evaluation

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 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.


 Fig. 6. Service time versus λ .

 Fig. 7. Service time versus τ .

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 optimal range

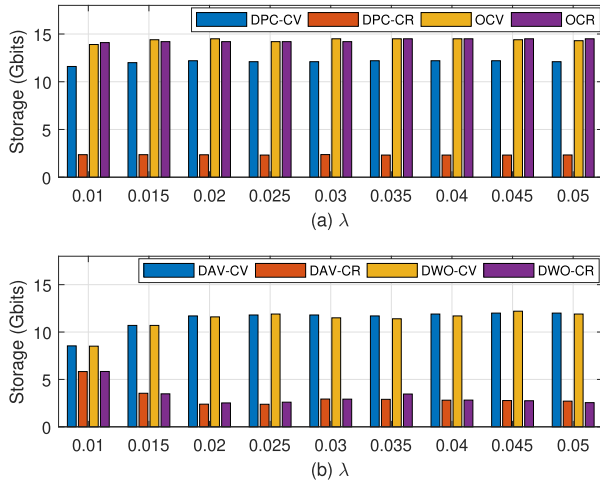
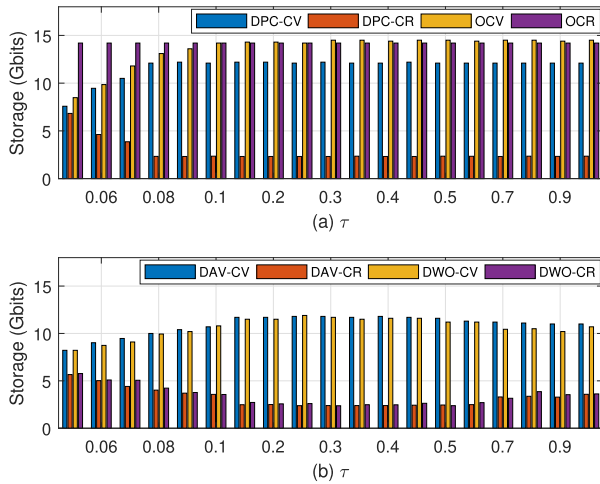

 Fig. 8. Storage consumption versus α .

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 (DPC-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.

Fig. 9. Storage consumption versus λ .Fig. 10. Storage consumption versus τ .

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 et al., "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, Thirdquarter 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, Jun. 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. 10291–10305, Sep. 2020.
- [5] H. Tian et al., "CoPace: Edge computation offloading and caching for self-driving with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13281–13293, 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. 13196–13212, 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. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9786–9796, Jul. 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. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7422–7434, Jul. 2023.
- [11] X. Zhou et al., "Edge computation offloading with content caching in 6G-enabled IoV," *IEEE Trans. Intell. Transp. 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 Comput.*, 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 Eng. J.*, vol. 126, pp. 555–564, Jul. 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 et al., "Real-time optimized clustering and caching for 6G satellite-UAV-terrestrial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 3009–3019, Mar. 2024.
- [16] Y. Zhao et al., "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 et al., "CAAF: An NDN-based cache-aware adaptive forwarding strategy for reliable content delivery in VANETs," *IEEE Trans. Mobile Comput.*, vol. 24, no. 7, pp. 6361–6375, Jul. 2025.
- [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. Sustain. Comput.*, vol. 9, no. 4, pp. 615–630, Jul./Aug. 2024.

- [19] Y.-T. Wang et al., "Markov clustering-based content placement in roadside-unit caching with deadline constraint," *IEEE Trans. Intell. Transport. Syst.*, vol. 25, no. 9, pp. 11881–11892, 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. Netw. Conf.*, 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, Jun. 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, Jul. 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, 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. Comput. Commun. Workshops*, Apr. 2016, pp. 314–319.
- [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. 12657–12667, 2017.
- [29] S. A. Elsayed, S. Abdelhamid, and H. S. Hassanein, "Probabilistic cooperative caching in VANETs for social networking," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2018, pp. 1–7.
- [30] G. Manzo et al., "DeepNDN: Opportunistic data replication and caching in support of vehicular named data," in *Proc. IEEE Int. Symp. A World Wireless, Mobile Multimedia Netw.*, 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 Comput.*, 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 Things J.*, vol. 10, no. 16, pp. 14834–14845, 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 Things J.*, vol. 8, no. 11, pp. 8964–8975, Jun. 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 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. Intell. Transp. 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. Intell. Transp. 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 Things J.*, vol. 11, no. 14, pp. 25198–25212, Jul. 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, 2020.
- [39] B. Bandyopadhyay, P. Kuila, and M. C. Govil, "Popularity-conscious service caching and offloading in digital twin and NOMA-aided connected autonomous vehicular systems," *IEEE Trans. Netw. Service Manage.*, 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. Sci. Eng.*, 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, Jun. 2018.
- [42] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Comput. 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. Consum. Electron.*, 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.*, vol. 74, no. 5, pp. 7084–7098, May 2025.
- [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. Transp. Syst.*, 2022, pp. 377–382.
- [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.*, 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. Comput. Commun.*, 1999, pp. 126–134.
- [52] S. Zhu et al., "Probability distribution of Rician K-factor in urban, suburban and rural areas using real-world captured data," *IEEE Trans. Antennas Propag.*, vol. 62, no. 7, pp. 3835–3839, Jul. 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.*, 2018, pp. 1–6.
- [54] B. Hu, L. Fang, X. Cheng, and L. Yang, "In-vehicle caching (IV-Cache) via dynamic distributed storage relay (D^2 SR) in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 843–855, Jan. 2019.
- [55] Q.-N. Tran et al., "D2D multi-hop multi-path communications in B5G networks: A survey on models, techniques, and applications," *EAI Endorsed Trans. Ind. Netw. 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). Sheffield, U.K.: University of Sheffield, 1994.
- [58] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, pp. 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. 160510–160520, 2021.
- [60] H. Li, S. Wu, J. Jiao, X. -H. Lin, N. Zhang, and Q. Zhang, "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, O. Chughtai, Y. Zhang, A. H. Alenezi, and N. Alqahtani, "Framework for optimized resource allocation in multi-user, multi-service, multi-device aerial networks," *IEEE Access*, vol. 12, pp. 54866–54878, 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 Process.*, 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, 2018.
- [64] H. Zhang, X. Cao, J. K. L. Ho, and T. W. S. Chow, "Object-level video advertising: An optimization framework," *IEEE Trans. Ind. Informat.*, 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.

Regular Article

Joint Caching and Hovering Minimized Service Time for Cooperative Video Streaming in UAV-assisted VANETs

Quynh-Anh Nguyen¹, Nguyen-Son Vo^{2,4}, Van-Ca Phan³, Dac-Binh Ha⁴, Minh-Phung Bui⁵, Long D. Nguyen⁶

¹ Vietnam Aviation Academy, Ho Chi Minh City, Vietnam

² Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, Vietnam

³ Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam

⁴ Faculty of Electrical-Electronic Engineering, Duy Tan University, Da Nang, Vietnam

⁵ Faculty of Information Technology, School of Technology, Van Lang University, Ho Chi Minh City, Vietnam

⁶ Dong Nai University, Dong Nai, Vietnam

Correspondence: Nguyen-Son Vo, vonguyenson@duytan.edu.vn

Communication: received 26 December 2025, revised 27 January 2026, accepted 08 February 2026

Online publication: 09 February 2026, Digital Object Identifier: 10.21553/rev-jec.430

Abstract– The rapid growth of advanced applications and services (A&Ss), particularly video streaming, imposes stringent latency and quality-of-experience requirements on vehicular ad hoc networks (VANETs). Although edge caching at roadside units (RSUs) can reduce service time, its effectiveness is limited in highly dynamic environments characterized by uneven vehicular user (VU) distributions, constrained caching resources, and sparse roadside infrastructure. To address these challenges, this paper proposes a caching and hovering (CAHO) optimization design for cooperative video streaming in VANETs assisted by unmanned aerial vehicles (UAVs). In the proposed CAHO design, VU distributions are modeled as independently thinned one-dimensional homogeneous Poisson point processes, and RSU-to-VU and UAV-to-VU communication models together with video popularity characteristics are incorporated. Based on this modeling, the CAHO problem jointly optimizes video caching at RSUs and UAVs as well as UAV hovering positions to minimize the service time under limited storage resources. The CAHO optimization problem is efficiently solved using genetic algorithms combined with a penalty function and a divide-and-conquer strategy. Simulation results demonstrate that the proposed CAHO scheme outperforms benchmark schemes under various system scenarios in terms of service time and storage resource utilization, highlighting the effectiveness and practical potential for video streaming A&Ss in UAV-assisted VANETs.

Keywords– Edge caching, unmanned aerial vehicles, UAV-assisted VANETs, vehicular ad-hoc networks, video streaming applications and services.

1 INTRODUCTION

The proliferation of advanced applications and services (A&Ss) imposes stringent requirements for wireless network infrastructures, posing particular challenges to intelligent transportation systems based on vehicular ad hoc networks (VANETs). The accelerated development of the Internet of Things (IoT), autonomous vehicles, and smart devices, together with multimedia A&Ss, has driven mobile data traffic to exhibit exponential expansion. A statistical report in [1] estimates that approximately 300 million A&Ss will serve 12.3 billion mobile users (MUs), among which video traffic accounts for a dominant share, contributing up to 79% of mobile data traffic [2]. Consequently, the demand for enhanced quality of experience (QoE) from the MUs has grown substantially.

In this context, integrating VANETs with edge techniques emerges as a promising approach to bringing contents and computational resources closer to vehicular users (VUs), thereby mitigating the core network

load and reducing the end-to-end latency. Efficient edge caching-enabled techniques play a crucial role in supporting latency-sensitive services, e.g., video streaming A&Ss, by storing popular video contents at the network edge to simultaneously lower service latency and energy consumption [3]. Another study in [4] considers a highly dynamic environment characterized by uneven and dense distribution of VUs, where caching and offloading techniques deployed at roadside units (RSUs) struggle to guarantee quality of service (QoS) under massive and diverse content demands. To address this challenge, the authors designed a joint computation offloading and content caching framework to minimize the overall network delay while satisfying the long-term energy constraints.

For more efficient edge techniques in VANETs, timely studies have proposed cooperative strategies that jointly exploit storage and communication resources at RSUs, macro base station (MBS), and VUs to reduce network congestion and latency [5–7]. Particularly, the authors in [5] proposed a caching scheme

for VANETs that leverages location-based and popular contents, where each content is cached either at both the macro base station (MBS) and RSUs, only at the MBS, or only at RSUs. The objective is to minimize overall transmission delay and service cost. Another work in [6] combines caching at RSUs with broadcast transmission scheduling and transmission rate adaptation to maximize the offloaded traffic. In [7], a joint deterministic and probabilistic caching strategy is proposed for deployment at both RSUs and VUs. Moreover, a cooperative communication framework enabling video delivery from RSUs and VUs to VUs is developed to achieve minimal service time while efficiently utilizing the storage resources of both RSUs and VUs.

Recently, unmanned aerial vehicles (UAVs) have been utilized as an effective augmenting component for VANETs due to their flexibility in selecting hovering positions with a high line-of-sight (LoS) probability, extended communication coverage, and rapid deployment. UAVs can act as aerial base stations to assist VANETs in delivering contents over areas where RSUs are overloaded or terrestrial infrastructure is insufficient [8, 9], [10–12]. In line with this approach, several studies [13–15] have exploited UAVs to design routing protocols and temporal data dissemination algorithms for mitigating post-disaster communication disruptions and delay-sensitive constraints in VANETs. When caching considered, the work [16] deploys UAVs with limited bandwidth resources to assist VANETs. The proposed scheme facilitates content dissemination among VUs through network coding-based cooperative caching to improve bandwidth efficiency. In [17], a UAV is employed to cache content and serve dynamic VUs, with joint trajectory planning and cache management considered to improve the data download amount and energy efficiency. Furthermore, caching can be performed at both UAVs and VUs by jointly utilizing user preferences together with incentive-driven caching and replacement schemes [18]. This solution effectively reduces the request latency and energy consumption.

Other UAV-assisted VANETs with caching have been investigated in the context of human-centric consumer applications, computation offloading, and task offloading [19–21]. However, existing studies in [19, 20] have not fully addressed the joint caching of contents at both RSUs and UAVs for cooperative video streaming A&Ss. Moreover, the determination of UAV hovering positions across diverse wireless propagation environments remains insufficiently investigated, particularly when minimizing service time under constrained caching storage resources. Although caching and replacement schemes at RSUs and UAVs are considered in [21], the hovering positions of UAVs are not taken into account, which limits the achievable delivery efficiency.

In this paper, by modeling the VUs using independently thinned one-dimensional homogeneous Poisson point process (T1D-PPP), incorporating RSU-to-VU (R2V) and UAV-to-VU (U2V) communication models, and capturing video popularity patterns, we propose a caching and hovering optimization (CAHO) design for cooperative video streaming A&Ss. In particular,

Table I
NOTATIONS

Notations	Specifications
M	Number of UAVs
N	Number of RSUs
K	Number of zones covered by each RSU
I	Number of videos
S_i	Size of video i , $i = 1, 2, \dots, I$
p_i	Popularity of video i
h_i	Caching radius of video i in RSUs
s_n	Average speed of VUs in road segment n covered by RU n , $n = 1, 2, \dots, N$
b_n	Hovering index, $b_n = 1$ if a UAV hovers at RSU n , otherwise $b_n = 0$
$c_{n,i}$	Caching index of video i in a UAV hovering at RSU n , $c_{n,i} = 1$ if a UAV caches video i and hovers at RSU n , otherwise $c_{n,i} = 0$
d	Distance of a road segment
d_k	Distance of zone k in a road segment
R_k	Transmission rate in zone k in a road segment
λ	Density of VUs
α	Skewed access rate (popularity) exponent among videos
W	System bandwidth
P	Transmission power of UAVs
σ^2	Additive white Gaussian noise power

a CAHO optimization problem is formulated and solved to determine which videos are cached at RSUs and which videos are cached at UAVs associated with their hovering positions to compensate for the caching gaps among RSUs. The objective is to minimize the service time and guarantee continuous video streaming sessions under limited caching storage resources at both RSUs and UAVs. The CAHO optimization problem is solved by using genetic algorithms (GA), combined with a penalty function method and a divide-and-conquer strategy applied to each chromosome-encoded individual. Simulation results show that the CAHO outperforms the other schemes across various system scenarios, thereby demonstrating the effectiveness and practical potential of the proposed solution for video streaming A&Ss in UAV-assisted VANETs.

The rest of this paper is organized as follows. We introduce the system model and describe how it works in Section II. Section III formulates the system model to derive the objective function of the CAHO optimization problem. The CAHO optimization design, i.e., CAHO optimization problem and its solution with GA, is presented in Section IV. In Section V, we evaluate the proposed CAHO method in comparison with other schemes. Finally, Section VI concludes the paper.

2 SYSTEM MODEL

In this paper, we deploy a CAHO model for cooperative video streaming in UAV-assisted VANETs as illustrated in Figure 1. The main notations used throughout the paper are summarized in Table I. Along the road, VUs

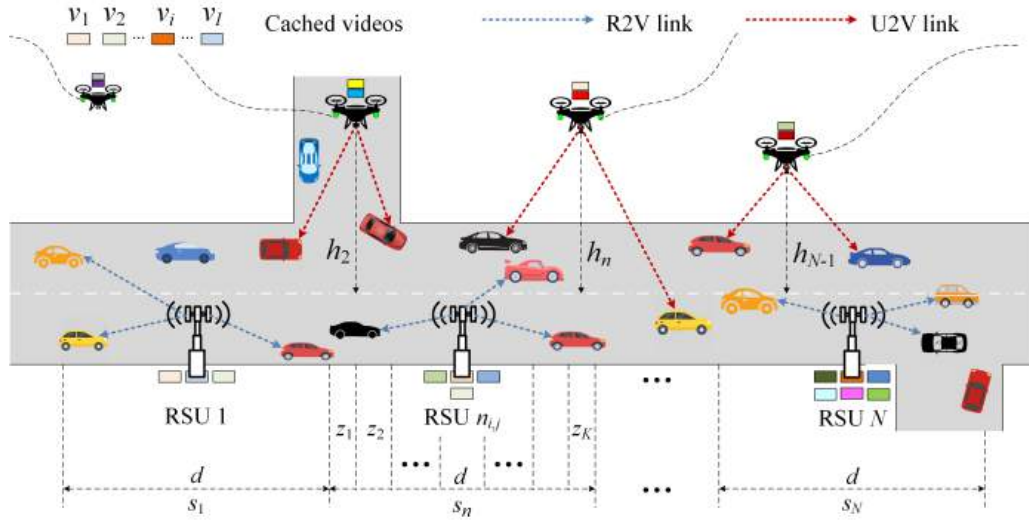


Figure 1. CAHO model for cooperative video streaming in UAV-assisted VANETs.

are spatially distributed according to T1D-PPP Φ with density λ [22, 23]. The road is divided into N segments, and the vehicle speed is assumed to be constant within each segment while varying across different segments, thereby capturing the realistic speed variations caused by traffic conditions, intersections, and driving behaviors. The model consists of N RSUs assisted by M UAVs for video streaming applications and services. There are I videos in the content library, where each video i is characterized by its popularity p_i and size S_i . Each RSU covers a segment of length d meters, which is further divided into K zones. The videos can be cached in both the RSUs and the UAVs. The VUs entering segment n , $n = 1, 2, \dots, N$, travel with an average speed of s_n kilometers per hour (km/h). Among the VUs, a subset is defined as requesting vehicular users (RVs) that request the videos.

In this context, to improve the QoS of video streaming A&Ss in VANETs, a CAHO optimization problem is formulated and solved for the optimal results of (i) caching placement (h_i) of video i in the RSUs, (ii) caching placement ($c_{n,i}$) of video i in a UAV hovering over segment n , and (iii) hovering index (b_n) over segment n at altitude h_n . The objective is to minimize the service time while efficiently utilizing the caching storage resources of RSUs and UAVs. The service time is defined as the average duration for the RVs to receive the requested videos from the RSUs and the UAVs while on the move. After deploying the CAHO method, when an RV requests video i , $i = 1, 2, \dots, I$, and enters a road segment, it is served by an RSU or a UAV that has cached video i to ensure a more continuous streaming session.

It is noted that if video i (i.e., of large size) cannot be completely transmitted after the RV traverses all N RSUs, the transmission will be continued by a subsequent set of RSUs associated with another MBS. In this work, under moderate network scales, a single MBS can reasonably act as a centralized controller that aggregates network information and disseminates the

optimization decisions to associated RSUs and UAVs. For large-scale systems with a high number of RSUs, UAVs, and videos, practical implementation indeed faces two main challenges: (i) the difficulty of collecting timely and accurate system information (e.g., channel state information), and (ii) the increased computational complexity of the optimization algorithms. In such scenarios, the number of RSUs, UAVs, and videos managed by each MBS should be appropriately constrained in accordance with the processing capability of the MBS. Moreover, for larger-scale and highly dynamic VANETs, the proposed CAHO model can be extended to a distributed or cooperative multi-MBS architecture, where multiple MBSs coordinate to serve vehicular users in a scalable manner, similar to the multi-MBS settings considered in prior studies such as [24]. This scenario is beyond the scope of this paper and is left for future investigation. The detailed system formulations for CAHO design and optimization are presented in the sequel.

3 SYSTEM FORMULATIONS

3.1 T1D-PPP Model for RVs

The use of the T1D-PPP model is primarily motivated by its analytical tractability and its ability to provide fundamental insights into the performance trends of large-scale vehicular networks [25, 26]. Several prior works have demonstrated that more complex vehicular models, such as Cox-based models with random road geometries, are equivalent to one-dimensional or two-dimensional PPP models under certain regimes of line density, vehicle density, or signal-to-interference ratio (SIR), particularly in the low-SIR region [27, 28]. In the considered UAV-assisted VANET with caching and video streaming, LoS-dominated links and relatively dense vehicular deployments lead to an interference-limited regime where the SIR thresholds of interest are moderate to low. Furthermore, VUs are associated with nearby caching entities, including road-side units

and UAVs, following proximity-based association rules. As a result, both the desired signal and the dominant interference originate from nearby transmitters along the same road, which aligns well with the assumptions underlying the T1D-PPP model. Given the T1D-PPP model for the spatial distribution of VUs on the one-way roads [22, 23], let p_i be the probability of video i , the distributions of RVs also follow the T1D-PPPs Φ_i^{RV} of density λ_i^{RV} , expressed as

$$\lambda_i^{\text{RV}} = p_i \lambda, \quad (1)$$

where the popularity video i modelled by a Zipf-like distribution [29] is given by

$$p_i = \frac{i^{-\alpha}}{\sum_{l=1}^L l^{-\alpha}}, \quad (2)$$

where $\alpha \geq 0$ denotes the coefficient characterizing the skewness of the video popularity distribution; for example, $\alpha = 0$ indicates that all videos have identical popularity.

3.2 Caching at RSUs and UAVs

3.2.1 Caching at RSUs: For caching in RSUs, we define h_i as the caching radius of video i to identify that video i is cached after every h_i hops in RSU $n_{i,j}$, which is computed as

$$n_{i,j} = h_i(j-1) + 1, j = 1, 2, \dots, J_i, \quad (3)$$

where J_i is the number of RSUs caching video i , given by

$$J_i = \left\lfloor \frac{N-1}{h_i} \right\rfloor + 1, \quad (4)$$

where the operator $\lfloor \cdot \rfloor$ is used to return the greatest integer less than or equal to a given real number. An example of caching radius $h_i = 5$ is illustrated in Figure 2.

3.2.2 Caching at UAVs: We can see that for particular video i , the number of RSUs that do not cache it is $(N - J_i)$, namely, caching gaps. It is challenging to deploy a large number of UAVs just for caching video i to fulfil the caching gaps. So, given M UAVs, each UAV has to selectively cache a number of videos and hovers above a proper RSU which does not cache them to ensure continuous video streaming sessions (Figure 2). For caching in UAVs, a UAV hovers above RSU $n \neq n_{i,j}$ to transmit video i to the associated RVs if $c_{n,i} = 1$, otherwise $c_{n,i} = 0$.

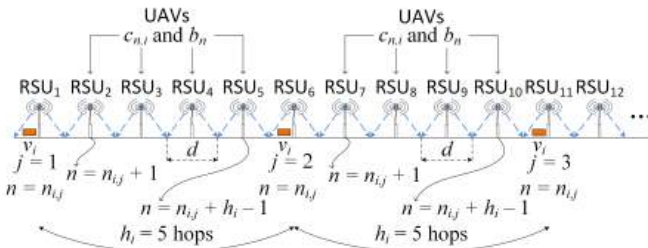


Figure 2. An example of caching placement and hovering with $h_i = 5$.

Table II
ZONES COVERED BY AN RU

Zones	1	2	3	4	5	6	7
$d_k(\text{m})$	25	30	40	60	40	30	25
$r_k(\text{m})$	125	100	70	30	70	100	125
$R_k(\text{Mbps})$	1	2	5.5	11	5.5	2	1

3.3 R2V Transmission Capacity

To derive the R2V transmission capacity, we apply the R2V channel model with WiFi access technology given in [5, 30]. In this model, the range d of a road segment covered by an RSU is divided into K zones. The parameters of zone k , $k = 1, 2, \dots, K$, $K = 7$, are listed in Table II. The capacity per RV requesting video i in zone k served by RSU n is given by

$$C_{n,k,i}^{\text{R2V}} = \begin{cases} \frac{p_{k,i}^{\text{RV}} R_k}{V_{k,i}}, & \text{if } n = n_{i,j}, \\ 0, & \text{if } n \neq n_{i,j}, \end{cases} \quad (5)$$

where $p_{k,i}^{\text{RV}}$ and $V_{k,i}$ denote, respectively, the probability that at least one RV in zone k requests video i from RSU n and the number of RVs requesting video i within zone k , expressed as

$$p_{k,i}^{\text{RV}} = 1 - e^{-d_k \lambda_i^{\text{RV}}}, \quad (6)$$

and

$$V_{k,i} = d_k \lambda_i^{\text{RV}}. \quad (7)$$

3.4 U2V Transmission Capacity

At RSU $n \neq n_{i,j}$, a UAV hovers above the centerline of the road at altitude h_n and transmits video i to the RVs by following the air-to-ground probabilistic path loss channel [12, 31–33]. We assume that each UAV, which is treated as a common cellular user, is allocated an orthogonal downlink spectrum resource for transmission. Moreover, due to the favorable LoS propagation of U2V links, the impact of interference can be neglected. Under this assumption, mutual interference among U2V links is further mitigated, and thus is not considered in this work. This channel, which is shown in Figure 3, consists of LoS and non-LoS (NLoS) links. The path loss of LoS and NLoS links between a UAV and a typical RV in zone k belonging to RSU n is given by

$$\psi_{n,k}^{\mathcal{A}} = \left(\frac{4\pi f_c d_{n,k}}{c} \right)^2 \eta_n^{\mathcal{A}}, \mathcal{A} \in \{\text{LoS}, \text{NLoS}\}, \quad (8)$$

where f_c is the carrier frequency, c is the speed of light, $\eta_n^{\mathcal{A}}$ is the attenuation factor of LoS link or NLoS link, and similar to the R2V channel model, the worst channel with the longest distance $d_{n,k}$ from the UAV to the RVs remains unchanged when moving in zone k as shown in Figure 3.

By further considering the occurrence probabilities of the LoS link ($p_{n,k}^{\text{LoS}}$) and of the NLoS link ($p_{n,k}^{\text{NLoS}} = 1 - p_{n,k}^{\text{LoS}}$), we can derive the following average

problem is formulated as

$$\min_{h_i, c_{n,i}, b_n} \bar{T}, \quad (15a)$$

$$\text{s.t. } 1 \leq h_i \leq N-1, \forall i, \quad (15b)$$

$$S^{\text{RSU}} + S^{\text{UAV}} \leq \delta N \sum_{i=1}^I S_i, \quad (15c)$$

$$\sum_{n=1}^N b_n \leq M, \quad (15d)$$

where (15c) is used to limit the total storage consumption for caching in RSUs and UAVs, and S^{RSU} and S^{UAV} are the caching storage resources consumed by RSUs and UAVs, respectively computed as

$$S^{\text{RSU}} = \sum_{i=1}^I J_i S_i, \quad (16)$$

and

$$S^{\text{UAV}} = \sum_{n=1}^N \sum_{i=1}^I c_{n,i} b_n S_i. \quad (17)$$

4.2 GA Solution

GA has been widely applied as an effective solution for complicated optimization problems in a broad range of real-world domains [35], spanning from communication systems [36–38] to video streaming A&Ss [39–42]. The effectiveness of GA derives from the evolutionary principles inspired by natural selection and genetic variation. Leveraging these principles, GA can flexibly search for exact or near-global optimal solutions in both unimodal and multimodal search spaces. By exploring multiple peaks of the search space in parallel, GA can avoid local convergence and thus enhance the likelihood of reaching globally optimal solutions [39, 40]. Owing to its robustness, adaptability, and strong global search capability, GA has been extensively used to overcome the challenges of practical optimization problems.

In this paper, we apply GA [43] to solve (15). However, the problem is that GA supports only simple constraints in the form of lower and upper bounds, such as (15b), but not more complex ones like (15c) and (15d). This problem is addressed by using the penalty method [39]. To this end, we convert (15) into an unconstrained optimization problem by reformulating (15c) and (15d) as

$$\begin{cases} \Delta S = \delta N \sum_{i=1}^I S_i - S^{\text{RSU}} - S^{\text{UAV}} \geq 0, \\ \Delta M = M - \sum_{n=1}^N b_n \geq 0. \end{cases} \quad (18)$$

Then, we derive the penalty function as

$$F = \rho_1 (\min\{0, \Delta S\})^2 + \rho_2 (\min\{0, \Delta M\})^2, \quad (19)$$

where ρ_1 and ρ_2 are the constraint violation degrees used to adjust the degree of punishment if the individuals in GA violate the constraints.

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

$$\min_{h_i, c_{n,i}, b_n} T_F = \bar{T} + F. \quad (20)$$

The detailed GA used to solve (20) is presented in Algorithm 2. In Algorithm 2, the two integer and binary

Algorithm 2 GA for CAHO.

Require: System and GA parameters in Tables III and Table IV
 $Gen = 1$: Generation count
Ensure: $T_F^*(h_i^*, c_{n,i}^*, b_n^*)$

- 1: Randomly generate
 - 1) N_P sub-strings $\{X_1^{(z)}\}$, $z = 1, 2, \dots, N_P$, each of I integer optimization variables $\{h_i\}$.
 - 2) N_P sub-strings $\{X_2^{(z)}\}$, each of $(N \times I + N)$ bits to characterize a set of $N \times I$ binary optimization variables $\{c_{n,i}\}$ and N binary optimization variables $\{b_n\}$.
- 2: Compute N_P fitness values based on (20) to have $\{T_F^{(z)}\}$, i.e., $T_F^{(z)}(X_1^{(z)}, X_2^{(z)})$.
- 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 $N_{PG} = N_P \times P_G$ best individuals with lowest fitness values by using stochastic universal sampling operator [43] for breeding to obtain $\{X^{(t)}\} = [\{X_1^{(t)}\}\{X_2^{(t)}\}]$, $t = 1, 2, \dots, N_{PG}$.
- 6: Choose a pair of parents to create the offsprings by using single point crossover with probabilities P_{ci} and P_{cb} 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_{mb} 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 $T_F^{(t),*}(X_1^{(t),*}, X_2^{(t),*})$.
- 9: Reinsert $\{X^{(t),*}\}$ and $T_F^{(t),*}$ into the present generation to obtain the new sets of $\{X^{(z)}\}$ and $T_F^{(z)}$.
 $Gen = Gen + 1$
- 11: **end while**
- 12: Find the best fitness $T_F^*(h_i^*, c_{n,i}^*, b_n^*) \in \{T_F^{(z)}(X_1^{(z)}, X_2^{(z)})\}$.

optimization variables (OVs) 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 [43] is used to generate the integers from 0 to $(N-2)$, and then added by 1 to satisfy the constraint (15b). 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 derived from (19) is less than 10^{-3} in 10 consecutive generations.
- Generation count (Gen) is equal to a given number of generations ($N_G = 100$).

As an adaptive heuristic search algorithm, the time and memory complexity of Algorithm 2 is primarily determined by the population size N_P and the number of generations N_G . The values of N_P and N_G may vary depending on several factors, including: 1) the system scale characterized by M , N and I ; 2) the computational burden associated with the objective function, constraints, and optimization variables in (15); and 3) the accuracy requirements of video streaming A&Ss.

Table III
SYSTEM AND GA PARAMETERS

Symbols	Specifications
M	5 UAVs
N	25 RSUs
K	7 zones covered by each RSU
I	10 videos
S_i	Uniformly random distributed from 0.25 to 2.5 (Gbit)
s_n	Uniformly random distributed from 30 to 60 (km/h)
d	250 m
λ	0.025 VUs/m
α	1
δ	0.5
W	10 MHz
P	5 W
σ^2	10^{-12} W
c	3×10^8 m/s
f_c	2 GHz
$\{\rho_1, \rho_2\}$	$\{10^{-6}, 1\}$, constraint violation degree properly determined based on the approach proposed in [40]
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
$\{P_{ci}, P_{cb}\}$	$\{0.9, 0.6\}$, set to sufficiently high values to ensure effective crossover and convergence
$\{P_{mi}, P_{mb}\}$	$\{10^{-12}, 10^{-12}\}$, set to sufficiently low values to ensure effective mutation and convergence

Table IV
PROPAGATION FEATURES [11, 34]

Parameters	κ_n	ν_n	η_n^{LoS}	η_n^{NLoS}	θ_n
Environments					
Suburban	4.88	0.43	0.1	21	20.34°
Urban	9.61	0.16	1	20	42.44°
Dense urban	12.08	0.11	1.6	23	54.62°
High-rise urban	27.23	0.08	2.3	34	75.52°

Table V
OVs OF DIFFERENT SCHEMES

Schemes	h_i	$c_{n,i}$	b_n
RAHO	✓	✓	✗
RACR	✗	✓	✓
RACU	✓	✗	✓
NUAV	✓	✗	✗
CAHO	✓	✓	✓

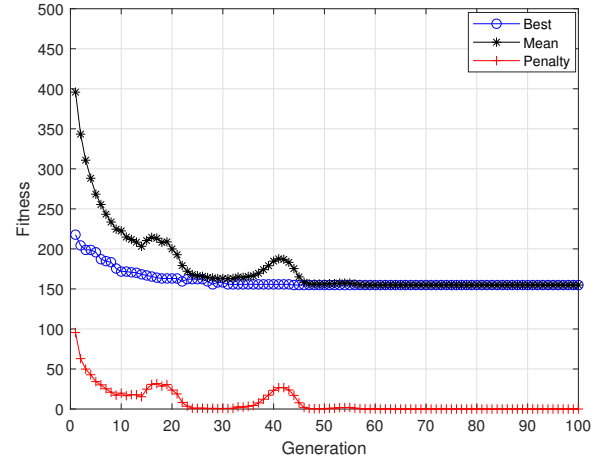


Figure 4. Convergence rate of GA.

Moreover, the overall complexity is also affected by the implementation of genetic operators such as selection, crossover, and mutation. Consequently, the complexity of Algorithm 2 is on the order of $\mathcal{O}(N_P N_G)$ [36, 40].

In practice, Algorithm 2 is more suitably executed at a single MBS, which centrally implements the CAHO strategy for all RSUs, UAVs, and popular videos within a limited road. Nevertheless, when the system scales up with longer road consisting of larger values of M , N , and I , practical deployment of CAHO faces two major challenges. On the one hand, acquiring the system information, such as channel state information, for formulating the CAHO optimization problem becomes increasingly difficult. On the other hand, the complexity of Algorithm 2 grows substantially. To overcome these challenges, the number of RSUs, UAVs, and popular videos assigned to each MBS should be carefully selected in accordance with the system scale and the processing capability of the MBS. Moreover, for scenarios involving large-size videos, multiple MBSs can collaboratively provide continuous video streaming A&Ss to RVs [24].

5 PERFORMANCE EVALUATION

5.1 Parameters Setting

The system and GA parameters are detailed in Table III. The propagation features of communication

environments in different segments, which are randomly selected, are shown in Table IV. To evaluate the performance of CAHO method, we compare it with the other four schemes, including (i) random hovering (RAHO), (ii) random caching at RSUs (RACR), (iii) random caching at UAVs (RACU), and (iv) none-UAV deployment (NUAV), with different considered optimization variables as listed in Table V. In RAHO, h_i and $c_{n,i}$ are optimized, but b_n is randomly generated. In RACR, $c_{n,i}$ and b_n are optimized, but h_i is randomly generated. In RACU, h_i and b_n are optimized, but $c_{n,i}$ is randomly generated. And in NUAV, no UAVs are deployed ($M = 0$). Obviously, the constraints in all RAHO, RACR, RACU, and NUAV have to be satisfied. Furthermore, we force the RAHO, RACR, RACU, and NUAV to cache as much storage as or (a little bit higher storage than) the CAHO method does to ensure a fair comparison.

5.2 GA Evaluation

Figure 4 illustrates the convergence behavior of GA in terms of the best fitness value (Best), the mean fitness value (Mean) defined in (20), and the average penalty

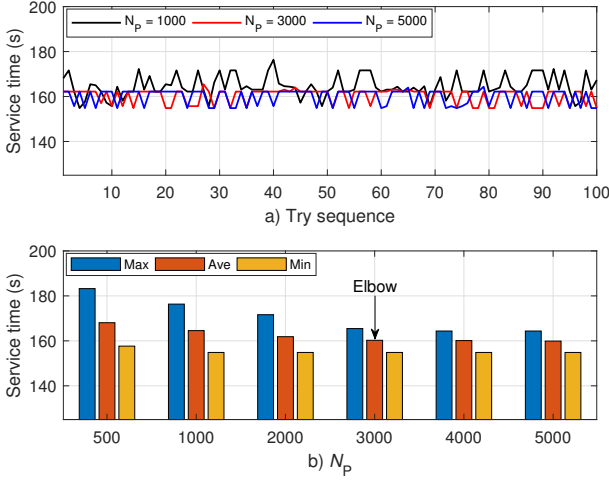
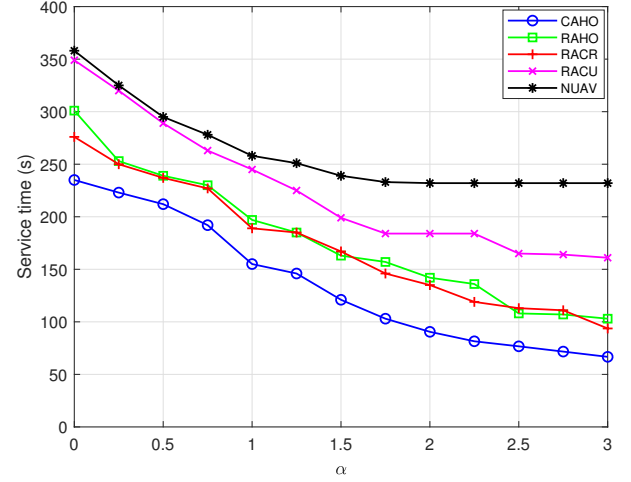
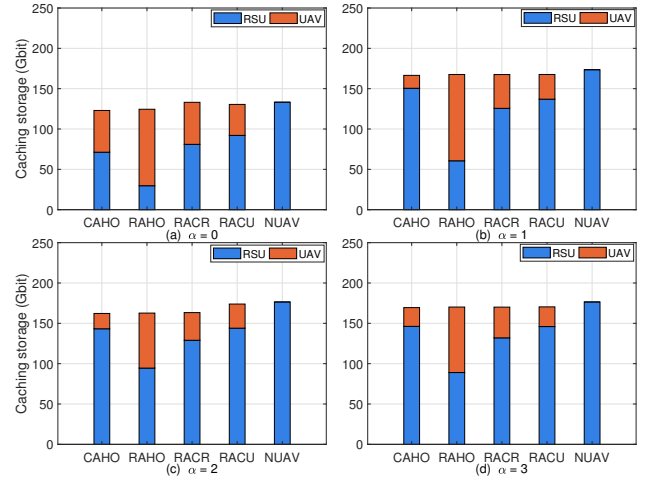


Figure 5. Convergence rate of GA.

value (Penalty) defined in (19) across all individuals, versus the number of generations N_G . In the early generations, the large gap between the Best and Mean values indicates a high population diversity, reflecting that GA is strong capability to effectively explore the searching space. As N_G increases, both the Best and Mean values gradually decrease and approach each other, demonstrating the convergence process of GA. Although the Best and Mean values become sufficiently close from around the 30th generation, the GA continues to further minimize the fitness function while driving the penalty term to zero, thereby ensuring that all constraints of the CAHO optimization problem are satisfied. After about 45 generations, the Mean converges to the Best and remains stable until the final generation, indicating that the GA reaches an optimal or approximately optimal solution. In addition, the rapid decrease of the Penalty toward zero confirms the feasibility of the final solution with respect to the system constraints.

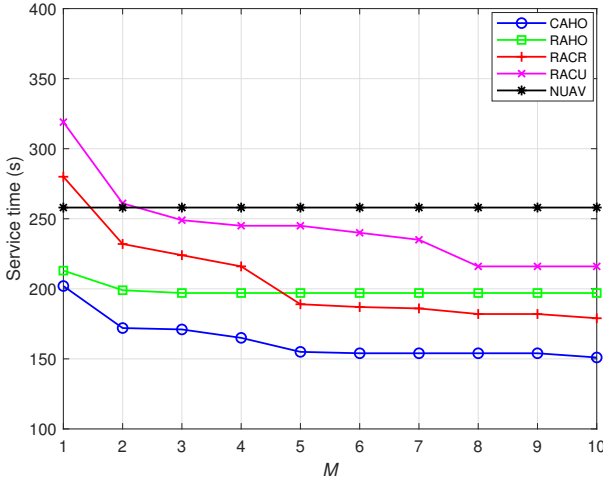
Besides the aforementioned convergence behavior for identifying the optimal solution, the stability and accuracy of GA are validated by executing the algorithm 100 independent tries for each population size ($N_p = \{500, 1000, 2000, 3000, 4000, 5000\}$), as illustrated in Figure 5. In Figure 5(a), for a small population size ($N_p = 1000$), the service time becomes more unstable and frequently takes higher values compared to the other two population sizes ($N_p = 3000, 5000$). As N_p increases to 3000 and 5000, the service time demonstrates significantly improved stability, with more accurate minimum values observed. To investigate the accuracy of GA, we further compute the maximum (Max), average (Ave), and minimum (Min) service time values over 100 independent tries for different population sizes as shown in Figure 5(b). For $N_p = 500$, the number of individuals is too small such that it cannot yield any accurate minimum values. A larger population size leads to a higher occurrence of accurate minimum values within 100 tries, thereby reducing the gap between the Ave and the Min. The results show

Figure 6. Service time performance versus α .Figure 7. Caching storage performance versus α .

that this improvement is marginal when $N_p > 3000$. Increasing the population size beyond 3000 does not yield significant gains in service time, while incurring a rapidly increasing computational cost. The elbow point at $N_p = 3000$ indicates an appropriate population size, at which GA achieves high accuracy with a reasonable cost. These results confirm the feasibility and effectiveness of GA in practical scenarios, achieving a good balance between accuracy and computational cost.

5.3 CAHO Performance Evaluation

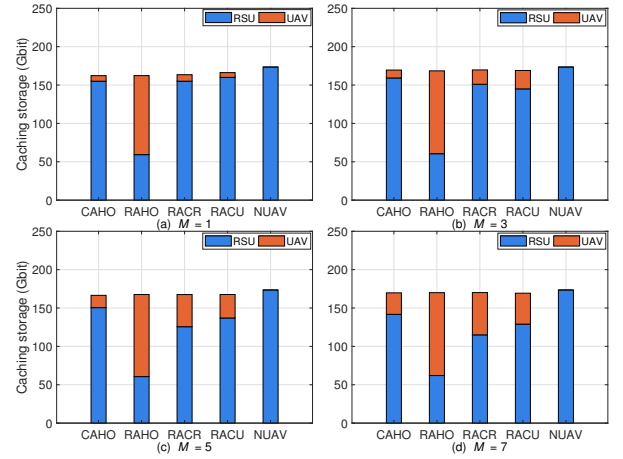
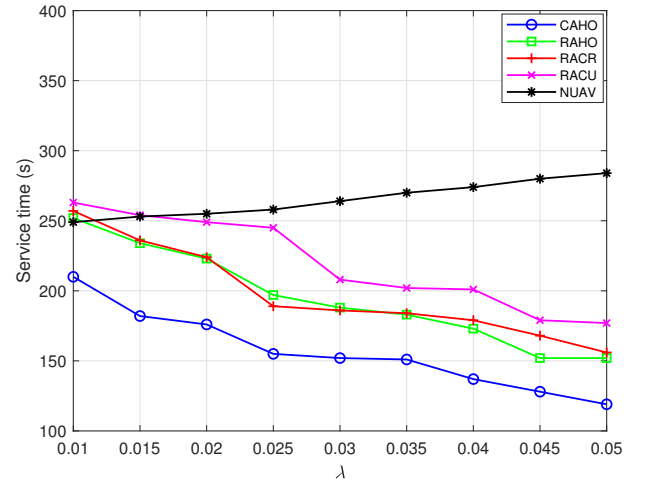
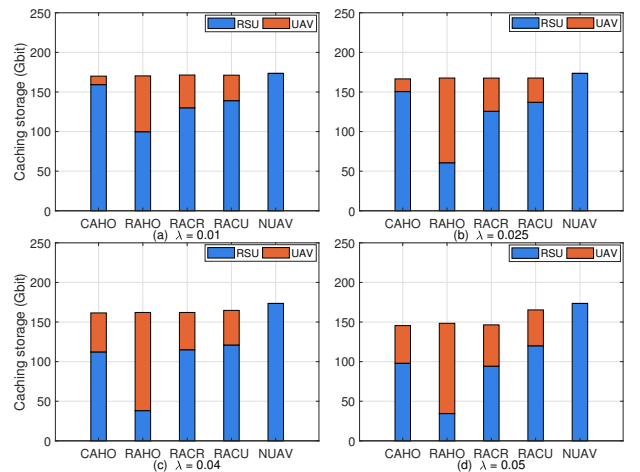
5.3.1 Performance versus α : Figure 6 plots the service time of different schemes versus α . As α increases, the service time of all schemes decreases, demonstrating the benefit of utilizing the caching resources for the most popular videos. CAHO achieves superior performance with the lowest service time by jointly optimizing all the three key OVs: caching radius at RSUs (h_i), caching placement at a UAV when it hovers over segment n ($c_{n,i}$), and hovering positions (b_n). Next to CAHO, RAHO and RACR provide comparable

Figure 8. Service time performance versus M .

performance, suggesting that UAV hovering and RSU caching play equivalent roles in service time reduction. From the performance of RACU, it can be observed that UAV caching is more critical than UAV hovering or RSU caching, as random UAV caching incurs a longer service time compared to random UAV hovering (RAHO) and random RSU caching (RACR). Finally, the assistance of UAVs contributes the most to system performance, since the NUAV scheme, which excludes both UAV hovering and UAV caching, yields the worst service time among all schemes.

Concerning caching storage consumption, Figure 7 plots the relationship between storage consumed by CAHO, RAHO, RACR, RACU, and NUAV versus α . The five schemes achieve similar total storage consumption amounts, in which CAHO provides a slightly lower level, ensuring the fairness in comparison. The main difference lies in how the total storage resource is allocated between RSUs and UAVs among the schemes. As α increases, UAVs equipped with hovering and caching tend to concentrate their caching on a smaller set of highly popular videos, thereby shifting storage usage toward RSUs rather than UAVs, except for the NUAV scheme. Among the schemes, CAHO utilizes the storage resource most efficiently. In contrast, RAHO suffers the lowest efficiency because the UAV hovering positions are randomly selected, which forces the UAVs to cache more videos trying to reduce the service time and compensate for the inefficiency in hovering.

5.3.2 Performance versus M : Figure 8 shows the service time of CAHO, RAHO, RACR, RACU, and NUAV versus the number of UAVs M . It is evident that the service time decreases as the number of UAVs increases for all UAV-assisted schemes. When M becomes large, further increasing the number of UAVs leads to service time saturation due to the limited storage resource and caching gaps. RAHO reaches saturation more quickly because the UAV hovering positions are not optimized, which hinders the effective utilization of a large number of UAVs for service time reduction compared with RACR. Since NUAV does not employ UAVs, its per-

Figure 9. Caching storage performance versus M .Figure 10. Service time performance versus λ .Figure 11. Caching storage performance versus λ .

formance remains unchanged with respect to M . Compared with other schemes, CAHO exploits additional UAVs more effectively, resulting in the lowest service

time. Importantly, the number of UAVs should be chosen carefully for each VANET scenario, ideally at the saturation elbow point (e.g., $M = 5$). The caching storage consumption of all schemes with respect to M is investigated in Figure 9, in which the behavior is similar to that in Figure 7. In addition, it is clear that as M increases, UAVs utilize more opportunities to achieve better U2V transmission capacity for filling the caching gaps, leading to an increase in the caching storage consumed at UAVs.

5.3.3 Performance versus λ : The service time performance of all schemes versus the density of VUs λ is presented in Figure 10. The results show that for UAV-assisted schemes, the service time generally decreases with increasing λ . However, NUAV exhibits the opposite behavior because expression (5), divided by $V_{k,i}$, increases as λ grows. A higher density of VUs increases the probability of having RVs and enables leveraging the advantages of U2V channels in UAV-assisted schemes, thereby compensating for the limitations of R2V channels in NUAV. Consequently, UAVs consume more caching storage resources as λ increases, as shown in Figure 11. Overall, the comparisons and analyses of system performance for all schemes with respect to α , M , and λ indicate similar behaviors, where CAHO consistently provides the RVs with the lowest service time and efficiently utilizes the caching storage resources.

6 CONCLUSION

In this paper, we have proposed the caching and hovering (CAHO) optimization design for cooperative video streaming in UAV-assisted VANETs. By jointly determining the cache placement at RSUs and UAVs along with UAV hovering positions, CAHO addresses the caching gaps and adapts to varying content popularity pattern, number of UAVs, and vehicle distributions. Comparative evaluations against the other benchmark schemes including RAHO, RACR, RACU, and NUAV, demonstrate that CAHO consistently achieves the lowest service time and efficiently balances the storage allocation between the RSUs and UAVs. GA employed to solve the CAHO optimization problem enables effective exploration of the joint optimization space and provides stable convergence behavior with high accuracy under appropriate population sizes, facilitating performance evaluation of the proposed framework. The results further highlight that UAVs are effectively utilized to complement RSU caching without overuse and can adapt well to system variations in latency-sensitive video streaming A&Ss in dynamic VANET environments. As a promising direction for future work, learning-based, lower-complexity and adaptive optimization algorithms will be investigated with insightful comparison to further capture the fast time-scale dynamics in highly mobile vehicular environments and to enhance system adaptability under rapidly changing network conditions.

ACKNOWLEDGMENT

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

REFERENCES

- [1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update," in *2017–2022 White Paper*, [Online]. Available: <https://www.cisco.com>, Feb. 2019.
- [2] Cisco, "Cisco annual internet report," in *2018–2023 White Paper*, [Online]. Available: <https://www.cisco.com>, Mar. 2020.
- [3] H. Feng, S. Guo, L. Yang, and Y. Yang, "Collaborative data caching and computation offloading for multi-service mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9408–9422, Sep. 2021.
- [4] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2212–2225, Apr. 2021.
- [5] J. Chen, H. Wu, P. Yang, F. Lyu, and X. Shen, "Cooperative edge caching with location-based and popular contents for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 291–10 305, Sep. 2020.
- [6] S. Berri, J. Zhang, B. Bensaou, and H. Labiod, "Joint content-prefetching, transmission scheduling, and rate adaptation in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4348–4358, Apr. 2022.
- [7] Q.-A. Nguyen, N.-S. Vo, T. C. Lam, T. Do-Duy, H. Jung, and T. Q. Duong, "Joint deterministic and probabilistic edge caching minimized service time for cooperative video transmission in VANETs," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 1932–1943, Sep. 2025.
- [8] P. Wu, X. Yuan, Y. Hu, and A. Schmeink, "UAV-enabled covert autonomous vehicular communication: Joint trajectory and resource allocation design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 12, pp. 21 769–21 783, Dec. 2025.
- [9] E. M. Mohamed, M. A. Alnakhli, and H. S. Hussein, "Contextual neural bandit approach for UAV selection in delay-doppler-based vehicular networks," *IEEE Wireless Communications Letters*, vol. 14, no. 11, pp. 3690–3694, Nov. 2025.
- [10] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2334–2360, Third-quarter 2019.
- [11] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.
- [12] A. Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proceedings of the 2014 IEEE Global Communications Conference*, Austin, TX, Dec. 2014, pp. 1–7.
- [13] Z. Fan, M. Zhang, Y. Cao, Z. Liu, O. Kaiwartya, and Y. Javed, "A novel UAV-assisted VANET routing protocol for post-disaster emergency communications," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 4863–4882, Dec. 2025.

- [14] Y. Shin, H.-S. Choi, Y. Nam, and E. Lee, "Particle swarm optimization-based content delivery protocol for UAV VANETs," *IEEE Access*, vol. 13, pp. 1594–1608, Dec. 2024.
- [15] X. Fan, H. Zhang, Y. Huang, Y. Su, H. Li, and J. Huo, "Temporal data dissemination in UAV-assisted VANETs through time-varying graphs," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 10, pp. 14 835–14 846, Oct. 2024.
- [16] K. Xiao, K. Feng, A. Dong, and Z. Mei, "Efficient data dissemination strategy for UAV in UAV-assisted VANETs," *IEEE Access*, vol. 11, pp. 40 809–40 819, Apr. 2023.
- [17] A. Al-Hilo *et al.*, "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, Aug. 2021.
- [18] G. Chen, Y. Han, F. Shen, Q. Zeng, and Y.-D. Zhang, "Edge collaborative caching based on incentive-driven D3QN combined with user preferences in UAV-assisted vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 12, pp. 22 945–22 961, Dec. 2025.
- [19] W. Wang, X. Xu, M. Bilal, M. Khan, and Y. Xing, "UAV-assisted content caching for human-centric consumer applications in IoV," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 927–938, Feb. 2024.
- [20] C. Li, K. Jiang, Z. Zhang, C. Xiong, and S. Wan, "Joint service caching and computation offloading scheme with 3D UAV deployment for ICVs in UAV-assisted VEC," *IEEE Transactions on Communications*, vol. 73, no. 11, pp. 10 886–10 899, Nov. 2025.
- [21] Y. Liu, C. Yang, X. Chen, and F. Wu, "Joint hybrid caching and replacement scheme for UAV-assisted vehicular edge computing networks," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 866–878, Jan. 2024.
- [22] J. P. Jeyaraj, M. Haenggi, A. H. Sakr, and H. Lu, "The transdimensional poisson process for vehicular network analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 8023–8038, Dec. 2021.
- [23] S. Zhao, Y. Zhang, Y. Zhu, Z. Zhao, and Y. Liu, "Optimal probabilistic collaborative caching in UAV-assisted vehicular networks," in *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China, Oct. 2022, pp. 1–6.
- [24] B. Feng, C. Feng, D. Feng, Y. Wu, and X.-G. Xia, "Proactive content caching scheme in urban vehicular networks," *IEEE Transactions on Communications*, vol. 71, no. 7, pp. 4165–4180, July 2023.
- [25] L. Jiao *et al.*, "Performance analysis of uplink/downlink decoupled access in cellular-V2X networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 5616–5630, May 2024.
- [26] K. Pandey, K. R. Perumalla, A. K. Gupta, and H. S. Dhillon, "Fundamentals of vehicular communication networks with vehicle platoons," *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 8634–8649, Dec. 2023.
- [27] V. V. Chetlur and H. S. Dhillon, "Success probability and area spectral efficiency of a VANET modeled as a Cox process," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 856–859, Oct. 2018.
- [28] V. V. Chetlur and H. S. Dhillon, "Coverage analysis of a vehicular network modeled as Cox process driven by poisson line process," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4401–4416, July 2018.
- [29] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proceedings of the IEEE INFOCOM'99. Conference on Computer Communications*, New York, NY, Mar. 1999, pp. 126–134.
- [30] W. Xu, W. Shi, F. Lyu, H. Zhou, N. Cheng, and X. Shen, "Throughput analysis of vehicular Internet access via roadside WiFi hotspot," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, p. 3980–3991, Apr. 2019.
- [31] W. Saad, M. Bennis, M. Mozaffari, and X. Lin, "Chapter 7 in *Wireless Communications and Networking for Unmanned Aerial Vehicles*", 1st ed. Cambridge University Press, 2020.
- [32] ITU-R, "Rec. p.1410-2 propagation data and prediction methods for the design of terrestrial broadband millimetric radio access systems," *Series, Radiowave propagation*, 2003.
- [33] M.-H. T. Nguyen *et al.*, "Spectrum-sharing UAV-assisted mission-critical communication: Learning-aided real-time optimisation," *IEEE Access*, vol. 9, pp. 11 622–11 632, Jan. 2021.
- [34] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 38–41, Feb. 2018.
- [35] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021.
- [36] 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.
- [37] H. Li *et al.*, "Energy-efficient task offloading of edge-aided maritime UAV systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1116–1126, Jan. 2023.
- [38] 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.
- [39] T. Fang and L. P. Chau, "GOP-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1323–1330, Jun. 2006.
- [40] 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.
- [41] H. Zhang, X. Cao, J. K. L. Ho, and T. W. S. Chow, "Object-level video advertising: An optimization framework," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 520–531, Apr. 2017.
- [42] G. Lai, F. F. Leymarie, and W. Latham, "On mixed-initiative content creation for video games," *IEEE Transactions on Games*, vol. 14, no. 4, pp. 543–557, Dec. 2022.
- [43] 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.



Quynh-Anh Nguyen received the B.E. degree in Electronics and Telecommunications from Ho Chi Minh City University of Transport, Vietnam, in 2008, and the M.Sc. degree in Mobile Communications from Télécom ParisTech, France, in 2012. She is currently pursuing the Ph.D. degree in Computer Science with the School of Computer Science and Artificial Intelligence, Duy Tan University, Vietnam. She is with the Faculty of Electrical and Electronics Engineering, Vietnam Aviation Academy, Vietnam. Her research interests include wireless communications, vehicular ad hoc networks (VANETs), UAV-assisted communications, and next-generation mobile networks.



interests include content caching and delivering in wireless networks, multimedia wireless communications, quality of experience, and sustainability provision in wireless networks for smart cities, and for disaster and environment management. He was a recipient of the Best Paper Award at the IEEE Global Communications Conference 2016 and the prestigious Newton Prize 2017.

Nguyen-Son Vo received the B.E. and M.E. degrees in electronics and telecommunications from Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam, in 2002 and 2005, respectively, and the Ph.D. degree in communication and information systems from the Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently with the Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City. His research



tional journals and conferences. Dr. Nguyen was awarded the Best Paper Award at the IEEE Digital Signal Processing (DSP) 2017, the IEEE International Conference on Recent Advances in Signal Processing, Telecommunication and Computing (SigTelCom) 2018, the IEEE International Conference on Communications (ICC) 2019, the International Wireless Communications & Mobile Computing Conference (IWCMC) 2019 and the IEEE Global Communications Conference (GLOBECOM) 2019, 2022. He was also awarded the Exemplary Reviewer Award in IEEE Communications Letters 2018 and 1st Prize "Technical Innovation" in the field of Information Technology - Electronics - Telecommunications, Dong Nai Province in 2023. His areas of interest in research of smart world include real-time convex/nonconvex optimization for real-world problem-solving (wireless communications, resource allocation, internet of things), digital transformation (data analysis, machine learning) modern computing (cloud computing, quantum computing) and embedded systems.

Long D. Nguyen [S'16, M'24] received his Ph.D. degree in Electronics and Electrical Engineering from Queen's University Belfast (QUB), UK, in 2018. He was a Research Fellow at Queen's University Belfast, UK for a part of Newton project (2018-2019). He is currently with the Department of Engineering, Dong Nai University as an Assistant Professor and Duy Tan University as an Adjunct Assistant Professor in Vietnam. He is serving as a reviewer for some IEEE Transactions, interna-



interests encompass wireless sensor networks, wireless and mobile communications, the Internet of Things (IoT), energy-efficient protocol design, and the integration of machine learning into wireless systems. Currently, his research focuses on cutting-edge projects in edge computing for video delivery systems in 6G networks and optimizing protocols for ultra-dense wireless environments, driving innovation in next-generation wireless communication technologies.

Van-Ca Phan received the B.E. and M.E. degrees in Electronics and Telecommunications from Ho Chi Minh City University of Technology, VNU-HCM, Vietnam, in 2002 and 2005, respectively, and the Ph.D. degree in Electronics and Radio Engineering from Kyung Hee University, South Korea, in 2011. He is currently an Associate Professor in the Faculty of Electrical and Electronics Engineering at Ho Chi Minh City University of Technology and Engineering, Vietnam. His research in-



networks, B5G/6G networks, mobile edge computing, quantum computing and communications.

Dac-Binh Ha received the B.S. degree in Radio Technique, the M.Sc. and Ph.D. degree in Communication and Information System from Huazhong University of Science and Technology (HUST), China in 1997, 2006, and 2009, respectively. He is currently Associate Professor of School of Engineering and Technology at Duy Tan University, Da Nang, Vietnam. His research interests are secrecy physical layer communications, cooperative communications, cognitive radio, RF energy harvesting



efficient routing in wireless sensor networks, multimedia wireless communication, and optimizing video storage and distribution systems.

Minh-Phung Bui received the Bachelor of Informatics degree from Van Lang University, Vietnam, in 2000, the M.S. degree in Computer Science from the University of Information Technology, VNU-HCM, in 2010, and the Ph.D. degree in Computer Science from Duy Tan University, Vietnam, in 2022. He is currently the Vice Dean of the Faculty of Information Technology at Van Lang University, Ho Chi Minh City, Vietnam. His research interests include UAV-assisted IoT networks, energy-