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

Nguyen-Son Vo[1*], Quynh-Anh Nguyen[2], Thanh-Minh Phan[3], Thanh-Hieu Nguyen[2], and Thanh-Dung Tran[4]

[1]Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, Vietnam
[2]Ho Chi Minh City University of Transport, Ho Chi Minh City, Vietnam
[3]Vietnam Aviation Academy, Ho Chi Minh City, Vietnam
[4]Absolute 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

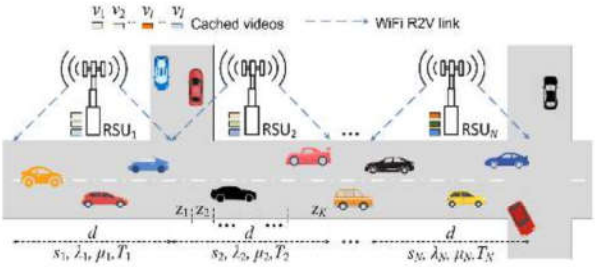
v_1 v_2 v_i v_I Cached videos
WiFi R2V link
RSU$_1$
RSU$_2$
RSU$_N$
z_1 z_2
z_K
d
$s_1, \lambda_1, \mu_1, T_1$
$s_2, \lambda_2, \mu_2, T_2$
$s_N, \lambda_N, \mu_N, T_N$

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,..., K
R_k	Transmission rate of zone k
S_i	Size of video i measured in bits, i = 1, 2,…, I
p_i	Access rate (popularity) of video i
λ_n	Average arrival rate of vehicles at RSU n, n = 1, 2,…, 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,..., 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,..., 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,..., 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

$$T_n = \frac{1}{\mu_n - \lambda_n}, \tag{1}$$

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

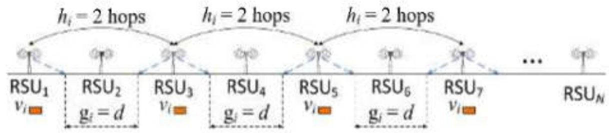

Figure 2. An example of the gap g_i between the RSUs in case $h_i = 2$.

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

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

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: $t_i$
for $j = 1 : J_i$ do
    $T_{W_i} = T_{W_i} + T_{n_{i,j}}$
    for $k = 1 : K$ do
        if $S_i > 0$ then
            $S_i = S_i - t_{k,n_{i,j}} R_{k,n_{i,j}}$
            $T_{X_i} = T_{X_i} + t_{k,n_{i,j}}$
        else
            $t_i = \min\left\{T_{X_i} + T_{W_i}, \frac{d}{t_{k,n_{i,j}}}\right\}$
            break
        end if
    end for
end for
```

$$\bar{D} = \sum_{i=1}^{I} p_i \sum_{n=1}^{\mathrm{N}} P_{i,n} D_{i,n}, \tag{3}$$

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

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

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

$$p_i = \frac{i^{-\alpha}}{\sum_{i=1}^{I} i^{-\alpha}}. \tag{5}$$

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,..., *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

$$R_{k,n} = \frac{R_k}{V_{k,n}}, \tag{6}$$

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

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

$$V_n = \lambda_n T_n \,. \tag{8}$$

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

$$t_{k,n} = \frac{d_k}{s_n} . \tag{9}$$

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

$$\overline{T} = \sum_{i=1}^{\mathrm{I}} p_i t_i \,. \tag{10}$$

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

$$S_T = \overline{D} + \overline{T} \,. \tag{11}$$

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

$$\min_{h_i} S_T \tag{12a}$$

$$\text{s.t. } 1 \le h_i \le N-1 \,, \forall i, \tag{12b}$$

$$\sum_{i=1}^{I} J_i S_i \le \delta N \sum_{i=1}^{I} S_i \,. \tag{12c}$$

where (12c) is used to limit the caching storage consumption and 0 < δ < 1 (if δ = 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

$$\Delta S = \delta N \sum_{i=1}^{I} S_i - \sum_{n=1}^{\mathrm{N}} J_i S_i \ \ge 0 \,. \tag{13}$$

We then derive the penalty function as

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

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

$$\min_{h_i} S_F = S_T + F \,. \tag{15}$$

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 $t = max\left\{\frac{d}{s_n}, n = 1, 2, \ldots, N\right\}$. 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

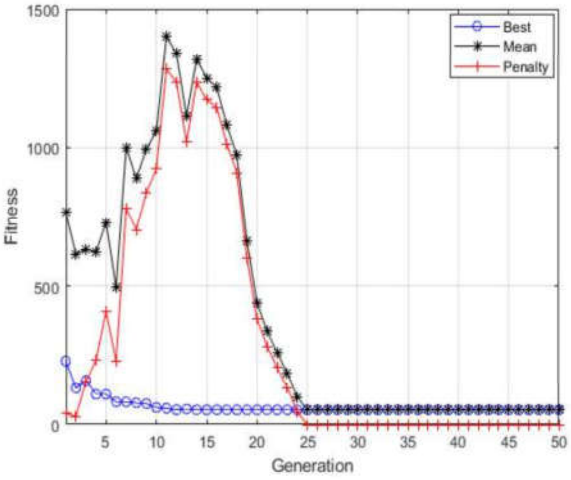

Figure 3. Convergence rate of GAs

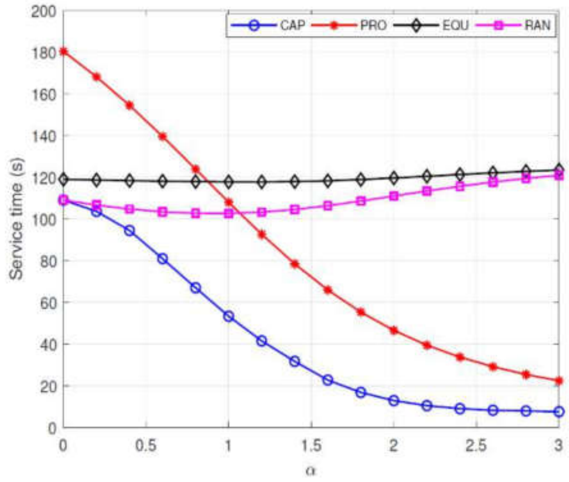

Figure 4. Service time versus *α*.

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 $h_i = \frac{\Omega}{\sqrt{r_i}}$, 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 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.

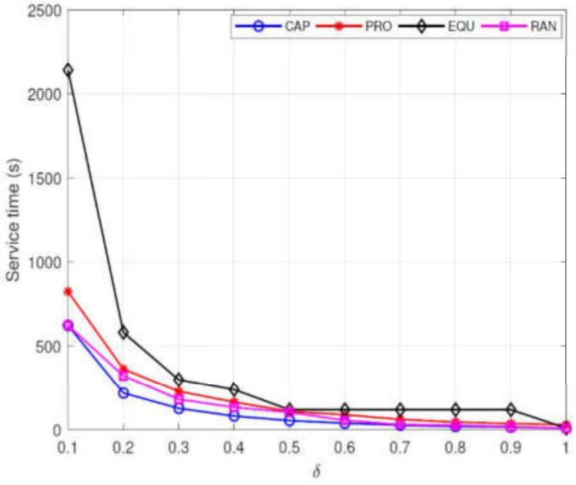

Figure 5. Service time versus δ

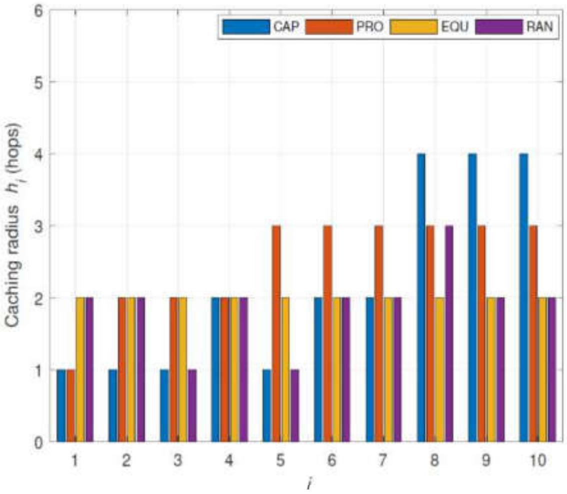

Figure 6. Caching radius h_i versus *i*.

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: Ajoint 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.