SCIENCE CHINA
Information Sciences

• Supplementary File •

Game-based computation offloading and resource allocation in stochastic geometry-modelling vehicular networks

Jianjie YANG[1], Zhijian LIN[1*], Yingyang CHEN[2], Xiaoqiang LU[1] & Yi FANG[3]

[1]*Department of Electronic and Information Engineering, Fuzhou University, Fuzhou* 350108*, China;*
[2]*College of Information Science and Technology, Jinan University, Guangzhou* 510632*, China;*
[3]*School of Information Engineering, Guangdong University of Technology, Guangzhou* 510006*, China*

Appendix A System Model

Appendix A.1 Distribution Model

In this scenario, there are a platoon of TVs driving on the coverage range of an RSU at a constant speed equipping with computing capacity. It is assumed that TVs obey the simplified vehicle-following model and the computing resources on each TV are the same.

TVs on the road are distributed stochastically, whose distribution practically obeys type-II Matérn hard-core point process (MHCPP) Φ_h which is a stochastic process considering the hard-core distance d_h between points based on one-dimensional Poisson point processes (1D PPP) Φ_v with density λ_v. Specifically, MHCPP Φ_h can be generated with its first-order density is given by [1]

$$\lambda_h = \frac{1-\exp\left(-2\lambda_v d_h\right)}{2d_h}. \tag{A1}$$

The RSU is laid on the roadside with a ground coverage radius R, and the distance from the middle of the road is S. Considering that the height of ENs is farther slighter than R, the height difference between ENs and RSU's foundation can be negligible. When TV m is driving in the coverage of the RSU and preparing to process a task, the initial distance from itself to the RSU is random, satisfying $d_m^{in} \in (S, R)$. At this time, there are two situations for d_m^{in}, shown in Fig. 1. In these cases, the corresponding staying time in the coverage of the RSU can be given by

$$t_{m,stay} = \begin{cases} \frac{\sqrt{R^2-S^2}+\sqrt{d_m^{in}-S^2}}{v}, & case1 \\ \frac{\sqrt{R^2-S^2}-\sqrt{d_m^{in}-S^2}}{v}, & case2 \end{cases}, \tag{A2}$$

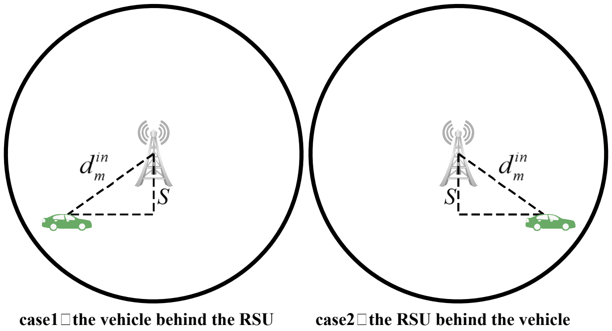
d_m^{in}
S
S
d_m^{in}
case1□the vehicle behind the RSU
case2□the RSU behind the vehicle

Figure A1 The situations of TV m staying in the RSU.

where v is the speed of TVs. It is assumed that the occurrence probabilities of both cases are the same. Considering the high-speed mobility of TVs, the distance between TV m and the RSU d_m is time-varying, which can be shown as

$$d_m\left(t\right) = \begin{cases} \sqrt{\left[\sqrt{(d_m^{in})^2-S^2}-vt\right]^2+S^2}, & case1 \\ \sqrt{\left[\sqrt{(d_m^{in})^2-S^2}+vt\right]^2+S^2}, & case2 \end{cases}. \tag{A3}$$

In the coverage of RSU, there are other trusted ENs in line of sight (LoS) with computation capacity distributed stochastically, including PCs, PVs, and passer-bys' MDs. Owing to the agglomeration of the same-type electronic devices, the distribution of ENs

[*] Corresponding author (email: zlin@fzu.edu.cn)

can be seen as a PCP approximately. It is assumed that ENs follow the Neyman-Scott process, the most common Poisson cluster process (PCP) in the area of spatial statistics [2]. The generation of PCP can be divided into two stochastic processes which are the parent and daughter process respectively, shown specifically as follows:

Parent process. The parent process is a PPP to ensure the crude locations of clusters. the clusters of ENs are distributed stochastically in the coverage of the RSU, following a PPP Φ_c with density λ_c. Accordingly, the probability that exists N clusters of ENs in the area centered on the RSU with radius R is

$$\mathbf{P}\left[\mathbf{N}_c\left(RSU, R\right) = N\right] = \frac{\left(\lambda_c \pi R^2\right)^N}{N!} \exp\left(-\lambda_c \pi R^2\right), \tag{A4}$$

where $\mathbf{N}_c\left(RSU, R\right)$ is the number of clusters obeying Φ_c in this area. According to (A4), the cumulative distribution function (CDF) of d_n, the distance from the RSU to nth closest cluster, can be calculated by

$$F_{d_n}\left(d\right) = P\left(d_n \leqslant d\right) = 1 - P\left(d_n > d\right) = 1 - \sum_{\varepsilon=0}^{n-1} \mathbf{P}\left[\mathbf{N}_c\left(RSU, d\right) = \varepsilon\right] = 1 - \sum_{\varepsilon=0}^{n-1} \frac{\left(\lambda_c \pi d^2\right)^{\varepsilon}}{\varepsilon!} \exp\left(-\lambda_c \pi d^2\right). \tag{A5}$$

Thus, the corresponding probability density function (PDF) is

$$\begin{aligned} f_{d_n}\left(d\right) =& \left[F_{d_n}\left(d\right)\right]' = \left[1 - \sum_{\varepsilon=0}^{n-1} \frac{\left(\lambda_c \pi d^2\right)^{\varepsilon}}{\varepsilon!} \exp\left(-\lambda_c \pi d^2\right)\right]' \\ =& 2\lambda_c \pi d \sum_{\varepsilon=0}^{n-1} \frac{\left(\lambda_c \pi d^2\right)^{\varepsilon}}{\varepsilon!} \exp\left(-\lambda_c \pi d^2\right) - 2\lambda_c \pi d \sum_{\varepsilon=1}^{n-1} \frac{\left(\lambda_c \pi d^2\right)^{\varepsilon-1}}{\left(\varepsilon-1\right)!} \exp\left(-\lambda_c \pi d^2\right) \\ =& 2\lambda_c \pi d \frac{\left(\lambda_c \pi d^2\right)^{n-1}}{\left(n-1\right)!} \exp\left(-\lambda_c \pi d^2\right). \end{aligned} \tag{A6}$$

Daughter process. The daughter process is to ensure the number and locations of ENs in each cluster. Assuming that the number of ENs in each cluster obeys a Poisson distribution with density λ_n, the probability that there are K_n ENs in cluster n is

$$\mathbf{P}\left(\mathbf{N}_{\widehat{n}} = K_n\right) = \frac{\left(\lambda_p\right)^{K_n}}{K_n!} \exp\left(-\lambda_p\right), \tag{A7}$$

where $\mathbf{N}_{\widehat{n}}$ is the number of ENs in cluster n. Accordingly, the set of ENs is $\mathcal{K} = \left\{\mathcal{K}_1, ..., \mathcal{K}_n, ..., \mathcal{K}_N \,|\, \mathcal{K}_n = \left\{1, ..., K_n\right\}, \forall n \in \mathcal{N}\right\}$. Because each cluster's coverage range is far smaller than the RSU's, the distances from the RSU to ENs within the same cluster can be approximated as equal.

Appendix A.2 Proof of Proposition 1

Particularly, the mmWave band is considered for V2R and R2N communications in this work and the V2R channel and R2N channel are orthogonal to each other. Generally, mmWave communications satisfy the Nakagami fading model, and the channel gain h satisfies gamma distribution, which can be given by [3]

$$f\left(h; \delta\right) = \frac{\delta^{\delta} h^{\delta-1}}{\Gamma\left(\delta\right)} \exp\left(-\delta h\right), \tag{A8}$$

where δ is a parameter representing the level of fading, and $\Gamma\left(\delta\right) = \int_0^{+\infty} z^{\delta-1} e^{-z} dz$.

In the scenario of the straight-line road, the blocks between TVs and the RSU can be ignored, and all the V2R links are in LoS. Thus, the corresponding path loss is [4]

$$\mathbf{PL}_{V2R}\left(d\right) = \left(\frac{4\pi d}{L_{mmw}}\right)^{-\alpha_R}, \tag{A9}$$

where L_{mmw} is the wavelength and α_R is the path fading exponent of V2R links. Thus, for the uplink between RSU and TV m, the received power is

$$P_{m,V2R}^{up}\left(t\right) = P_v h_m g_{ml} g_{ml} \mathbf{PL}_{V2R}\left[d_m\left(t\right)\right], \tag{A10}$$

where P_v is transmit power of TVs, h_m is the channel gain of this link and g_{ml} is the main-lobe gain of antennae. Meanwhile, interference from external communication links should be considered. Taking TV i for instance, the interference of the V2R uplink between itself and the RSU consists mainly of other V2R links, expressed as

$$I_{V2R} = \sum_{m \in \mathcal{M} \setminus i} h_m P_v G \mathbf{PL}_{V2R}\left[d_m\left(t\right)\right], \tag{A11}$$

where G is the discrete variable of the antenna gain based on the stochasticity of the antenna beam direction, whose values is shown as [5]

$$G \sim \left[\begin{array}{cccc} g_{ml} \cdot g_{ml} & g_{ml} \cdot g_{sl} & g_{sl} \cdot g_{ml} & g_{sl} \cdot g_{ml} \\ \frac{\theta}{\pi} \cdot \frac{\theta}{\pi} & \frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right) & \left(1 - \frac{\theta}{\pi}\right)\frac{\theta}{\pi} & \left(1 - \frac{\theta}{\pi}\right)\left(1 - \frac{\theta}{\pi}\right) \end{array}\right], \tag{A12}$$

where g_{sl} is the side lobe gain of antennae and θ is a half of the main lobe angle.

Theorem 1 (Campbell's Theorem [5])**.** Campbell's Theorem is to convert the expectation of a stochastic sum in the point process to an integral involving the point-process intensity function, which is specifically expressed as $\mathbb{E}\left(\sum\limits_{m\in\Phi}\mathcal{Z}\left(m\right)\right)=\int_{\mathbb{R}^2}\mathcal{Z}\left(x\right)\mathcal{H}\left(dx\right)=\int_{\mathbb{R}^2}\mathcal{Z}\left(x\right)\lambda\left(x\right)dx$, where Φ is a point process, $\mathbb{R}^2$ is the integral area of point process, and $\lambda\left(x\right)$ is the density function of the stochastic process.

According to Theorem 1, the average of I_{V2R} is given by

$$\begin{aligned}\overline{I}_{V2R}&=\mathbb{E}\left[\sum_{m\in\mathcal{M}\backslash i}h_mP_vG\mathbf{PL}_{V2R}\left(d_m\right)\right]\\&=P_v\overline{Gh_m}\int_{-\infty}^{+\infty}\lambda_h\mathbf{PL}_{V2R}\left[\sqrt{x^2+S^2}\right]dx=2\lambda_hP_v\overline{Gh_m}\int_0^{+\infty}\mathbf{PL}_{V2R}\left[\sqrt{x^2+S^2}\right]dx,\end{aligned}\tag{A13}$$

where $\overline{Gh_m}$ is the expectation of the antenna gain G and channel gain h_m. Thus, the transmit rate of the V2R uplink is

$$C_{m,V2R}^{up}\left(t\right)=\frac{B_{V2R}}{M}\log_2\left[1+\frac{P_{m,V2R}^{up}\left(t\right)}{\sigma^2+\overline{I}_{V2R}}\right],\tag{A14}$$

where B_{V2R} is the total bandwidth of V2R channels which is assumed to be uniformly distributed, and δ^2 is the noise power.

However, in the V2R downlink, the RSU occupies orthogonal frequency division multiplexing (OFDM) technology to return the task result so that the interference between different V2R links can be ignored [6]. Meanwhile, owing to the slight size of the returned result, the transmit time of the downlink is greatly short and the locations of TVs can be seen as constant during this time. Thus, the transmit time of the downlink can be expressed as

$$C_{m,V2R}^{down}=\frac{B_{V2R}}{M}\log_2\left[1+\frac{P_rh_mg_{ml}g_{ml}\mathbf{PL}_{V2R}\left[d_m\left(\Delta_m\right)\right]}{\sigma^2}\right],\tag{A15}$$

where $\Delta_m=\left(\vartheta_m^{RSU}+\sum\limits_{n=1}^{N}\vartheta_m^n\right)t_{m,V2R}^{up}+\vartheta_m^{RSU}t_{m,com}^{RSU}+\sum\limits_{n=1}^{N}\left[\vartheta_m^n\left(t_{m,R2N_n}^{down}+t_{m,com}^{n}+t_{m,R2N_n}^{up}\right)\right]$, and P_r is the transmit power of the RSU. Here, $t_{m,RSU}^{com}$ is the required time of calculation on the RSU, $t_{m,R2N_n}^{down}$ is the transmit time of the R2N downlink between the RSU and cluster n, $t_{m,n}^{com}$ is the required time of calculation on the EN in cluster n, and $t_{m,n}^{up}$ is the transmit time of the R2N uplink.

Appendix A.3 Proof of Proposition 2

The path fading of R2N links is

$$\mathbf{PL}_{R2N}\left(d\right)=\left(\frac{4\pi d}{L_{mmw}}\right)^{-\alpha_N},\tag{A16}$$

where α_N is path fading exponent of R2N links. The received power of an EN in cluster n from the RSU via the R2N downlink can be given by

$$P_{R2N_n}^{down}=P_rh_ng_{ml}g_{ml}\mathbf{PL}_{R2N}\left(d_n\right),\tag{A17}$$

where h_n is the channel gain of the links between the ENs in cluster n and the RSU.

Due to the stochasticity of the distribution of ENs, the method based on probability theory is introduced to compute the average transmit rate of the R2N downlink [7]. Owing to the application of OFDM, the interference between different R2N downlinks can be ignored. Thus, the coverage rate of the corresponding threshold of the signal-noise ratio (SNR) can be expressed as

$$\begin{aligned}\mathbf{P}_{SNR_{R2N_n}^{down}}\left(\chi\right)=&P\left(SNR_{R2N_n}^{down}>\chi\right)=P\left(\frac{P_{R2N_n}^{down}}{\sigma^2}>\chi\right)=P\left\{h>\chi\left[\frac{\sigma^2}{\Theta\left(d\right)}\right]\right\}\\=&\int_{\frac{\chi\sigma^2}{\Theta(d)}}^{+\infty}\frac{\delta^\delta h^{\delta-1}}{\Gamma\left(\delta\right)}\exp\left(-\delta h\right)dh\overset{(a)}{\approx}\left[\sum_{q=1}^{\delta}\frac{\left(\delta h\right)^{\delta-q}}{\left(\delta-q\right)!}\right]\exp\left(-\delta h\right)\Bigg|_{\chi\sigma^2/\Theta(d)}^{\infty}\\\overset{(b)}{\approx}&\mathbb{E}\left\{\left\{\sum_{q=1}^{\delta}\frac{\left[\frac{\delta\chi\sigma^2}{\Theta(d)}\right]^{\delta-q}}{\left(\delta-q\right)!}\right\}\exp\left[\frac{-\delta\chi\sigma^2}{\Theta\left(d\right)}\right]\right\}=\int_0^{\infty}\left\{\sum_{q=1}^{\delta}\frac{\left[\frac{\delta\chi\sigma^2}{\Theta(x)}\right]^{\delta-q}}{\left(\delta-q\right)!}\right\}\exp\left[\frac{-\delta\chi\sigma^2}{\Theta\left(x\right)}\right]f_{d_n}\left(x\right)dx,\end{aligned}\tag{A18}$$

where χ is threshold of SNR, (a) leverages the feature of the gamma function when $\delta\in\mathbb{Z}^+$,[1)] and (b) is to derive the average coverage ratio under different distances of the downlink. Accordingly, the coverage ratio of the corresponding transmit rate is

$$\begin{aligned}\mathbf{P}_{C_{R2N_n}^{down}}\left(\xi\right)=&P\left(C_{R2N_n}^{down}>\xi\right)=P\left[\left(B_{R2N}/\sum_{n=1}^{N}K_n\right)\log_2\left(1+SNR_{R2N_n}^{down}\right)>\xi\right]\\=&P\left(SNR_{R2N_n}^{down}>2^{\xi\sum\limits_{n=1}^{N}K_n/B_{R2N}}-1\right)=\mathbf{P}_{SNR_{R2N_n}^{down}}\left(2^{\xi\sum\limits_{n=1}^{N}K_n/B_{R2N}}-1\right).\end{aligned}\tag{A19}$$

where ξ is threshold of transmit rate. The average value of the corresponding transmit rate can be calculated by

$$\overline{C}_{R2N_n}^{down}=\mathbb{E}\left(C_{R2N_n}^{down}\right)=\int_0^{+\infty}\xi\mathbf{P}_{C_{R2N_n}^{down}}\left(\xi\right)d\xi\overset{(a)}{=}\int_0^{+\infty}\mathbf{P}_{SNR_{R2N_n}^{down}}\left(2^{\xi\sum\limits_{n=1}^{N}K_n/B_{R2N}}-1\right)d\xi,\tag{A20}$$

[1)] Generally, δ is set as an integer for convenience of researches.

where (a) is the application of positive random variables [8].

For the uplink, taking the EN in cluster j as an example, its main interference consists of uplink interference from ENs in other clusters and the RSU $I_{cl,R2N_j}^{up}$, and uplink interference from ENs in the same cluster $I_{in,R2N_j}^{up}$. Due to the slight distance between ENs in the same cluster, the interference inside the cluster can be simplified to $I_{in,R2N_j}^{up} = (K_j - 1)\, P_e \overline{Gh} \mathbf{PL}_{R2N}(d_j)$, where P_e is the transmit power of ENs. Hence, the total interference can be given by

$$I_{R2N_j}^{up} = I_{cl,R2N_j}^{up} + I_{in,R2N_j}^{up} = \sum_{n\in\mathcal{N}\backslash j} h_n P_e G \mathbf{PL}_{R2N}(d_n) + I_{in,R2N_j}^{up}, \tag{A21}$$

where h_n is the channel gain of links between ENs in cluster n and the RSU.

The coverage rate of signal-interference-noise ratio (SINR) of the R2N uplink between an EN in cluster j and the RSU can be expressed as

$$\begin{aligned}
\mathbf{P}_{SINR_{R2N_j}^{up}}(\chi) =& P\left(SINR_{R2N_j}^{up} > \chi\right) = P\left(\frac{P_{R2N_j}^{up}}{\sigma^2 + I_{R2N_j}^{up}} > \chi\right) = P\left\{h > \chi\left[\frac{\sigma^2 + I_{R2N_j}^{up}}{\Psi(d)}\right]\right\} \\
=& \int_{\chi\left[\frac{\sigma^2 + I_{R2N_j}^{up}}{\Psi(d)}\right]}^{+\infty} \frac{\delta^\delta h^{\delta-1}}{\Gamma(\delta)} \exp(-\delta h) dh \overset{(a)}{\approx} \left[\sum_{q=1}^{\delta} \frac{(\delta h)^{\delta-q}}{(\delta-q)!}\right] \exp(-\delta h)\Bigg|_{h=\chi\left(\sigma^2 + I_{R2N_j}^{up}\right)/\Psi(d)}^{\infty} \\
\overset{(b)}{\approx}& \mathbb{E}_{d_j}\left\{\left\{\sum_{q=1}^{\delta} \frac{\left\{\delta\chi\left[\frac{\sigma^2 + I_{R2N_j}^{up}}{\Psi(d)}\right]\right\}^{\delta-q}}{(\delta-q)!}\right\} \exp\left\{-\delta\chi\left[\frac{\sigma^2 + I_{R2N_j}^{up}}{\Psi(d)}\right]\right\}\right\} \\
=& \int_0^\infty \left\{\sum_{q=1}^{\delta} \frac{\left\{\delta\chi\left[\frac{\sigma^2+\Upsilon}{\Psi(x)}\right]\right\}^{\delta-q}}{(\delta-q)!}\right\} \exp\left\{\frac{-\delta\chi\left[\sigma^2 + \Xi(x)\right]}{\Psi(x)}\right\} \exp\left[\frac{-\delta\chi I_{cl,R2N_j}^{up}}{\Psi(x)}\right] f_{d_j}(x)\, dx \\
=& \int_0^\infty \left\{\sum_{q=1}^{\delta} \frac{\left\{\delta\chi\left[\frac{\sigma^2+\Upsilon}{\Psi(x)}\right]\right\}^{\delta-q}}{(\delta-q)!}\right\} \exp\left\{\frac{-\delta\chi\left[\sigma^2 + \Xi(x)\right]}{\Psi(x)}\right\} L_{I_{cl,R2N_j}^{up}}\left[\frac{-\delta\chi}{\Psi(x)}\right] f_{d_j}(x)\, dx.
\end{aligned} \tag{A22}$$

Here, the Laplace transform L_I of $I_{cl,R2N_j}^{up}$ can be calculated by

$$\begin{aligned}
L_I(s) =& \mathbb{E}\left(e^{-sI_{R2N_j}^{up}}\right) = \mathbb{E}\left\{\exp\left[-s\sum_{n\in\mathcal{N}\backslash j} h_n P_e G \mathbf{PL}_{R2N}(d_n)\right]\right\} = \mathbb{E}\left\{\prod_{n\in\mathcal{N}\backslash j} \exp\left[-sh_n P_e G \mathbf{PL}_{R2N}(d_n)\right]\right\} \\
=& \mathbb{E}\left\{\prod_{n\in\mathcal{N}\backslash j} \mathbb{E}_h\left\{\exp\left[-sh_n P_e G \mathbf{PL}_{R2N}(d_n)\right]\right\}\right\} \overset{(a)}{=} \mathbb{E}\left\{\prod_{n\in\mathcal{N}\backslash j}\left[1 + \frac{sP_e \overline{G} \mathbf{PL}_{R2N}(d_n)}{\delta}\right]^{-\delta}\right\} \\
\overset{(b)}{=}& \exp\left\{-2\pi\lambda_c\lambda_p \int_0^{+\infty}\left\{1 - \left[1 + \frac{sP_e \overline{G} \mathbf{PL}_{R2N}(y)}{\delta}\right]^{-\delta}\right\} dy\right\},
\end{aligned} \tag{A23}$$

where (a) is to calculate the interference from the R2N uplinks between ENs in a single cluster and the RSU when h follows a gamma distribution, and (b) utilizes the probability generation functional (PGFL) of PPP [9].[2)] Hence, the average probability can be derived as

$$\begin{aligned}
P_{C_{R2N_n}^{up}}(\xi) =& P\left(C_{R2N_n}^{up} > \xi\right) = P\left[\frac{B_{R2N}}{\sum\limits_{n=1}^{N} K_n} \log_2\left(1 + SINR_{R2N_n}^{up}\right) > \xi\right] \\
=& P\left(SINR_{R2N_n}^{up} > 2^{\xi\sum\limits_{n=1}^{N} K_n/B_{R2N}} - 1\right) = \mathbf{P}_{SINR_{R2Nn}^{up}}\left(2^{\xi\sum\limits_{n=1}^{N} K_n/B_{R2N}} - 1\right),
\end{aligned} \tag{A24}$$

The corresponding average transmit rate is

$$\overline{C}_{R2N_n}^{up} = \mathbb{E}\left(C_{R2N_n}^{up}\right) = \int_0^{+\infty} \xi P_{C_{R2N_n}^{up}}(\xi)\, d\xi = \int_0^{+\infty} P_{SINR_{R2N_n}^{up}}\left(2^{\xi\sum\limits_{n=1}^{N} K_n/B_{R2N}} - 1\right) d\xi, \tag{A25}$$

Appendix B Problem Analysis and Proposed Solution

Appendix B.1 Problem Formulation

The main objective is to reduce the energy consumption of the whole system, which includes computation and transmitted energy consumption.

[2)] The PGFL is to convert the expectation of stochastic multiplication of functions in the point process to an integral over the point-process domain, which is specifically expressed as $\mathbb{E}\left[\prod\limits_{m\in\Phi} \mathcal{Z}(m)\right] = \exp\left\{-\int_{\mathbb{R}^2}\left[1 - \mathcal{Z}(x)\right]\mathcal{H}(dx)\right\} = \exp\left\{-\int_{\mathbb{R}^2}\left[1 - \mathcal{Z}(x)\right]\lambda(x)\, dx\right\}$ if the point process obeys PPP.

For task Ω_m, the computation time of the partial task calculated locally is $t^{local}_{m,com} = l_m D_m F_m / f_{m,local}$. Its corresponding required energy consumption is $e^{local}_{m,com} = \kappa (f_{m,local})^3 t^{local}_{m,com}$, where κ is an energy consumption factor depending on hardware performance [10].

Considering the time variability, the required transmit time of partial Ω_m offloaded to the RSU $t^{up}_{m,V2R}$ has to satisfy

$$\int_0^{t^{up}_{m,V2R}} C^{up}_{m,V2R}(t)\, dt = o_m D_m. \tag{B1}$$

Additionally, the corresponding result-returned time is $t^{down}_{m,V2R} = \varsigma o_m D_m / C^{down}_{m,V2R}$, where ς is the ratio of the result and original task [11]. Generally, the total CPU cycle frequency of RSU will be allocated to tasks that require computation resources of the RSU for task process rather than be employed by any single task. For the part offloaded to the RSU of task Ω_m, if it is calculated by the RSU, the required time is $t^{RSU}_{m,com} = o_m D_m F_m / f_{m,RSU}$. The computation energy consumption of this part can be given by $e^{RSU}_{m,com} = \kappa (f_{m,RSU})^3 t^{RSU}_{m,com}$.

But if the partial task is reallocated to the EN in cluster n for computation offloading, the corresponding R2N downlink transmit time is $t^{down}_{m,R2N_n} = o_m D_m / \overline{C}^{down}_{R2N_n}$. Meanwhile, the corresponding returned transmit time can be calculated by $t^{down}_{m,R2N_n} = \varsigma o_m D_m / \overline{C}^{up}_{R2N_n}$. Besides, the required computation time is $t^{n}_{m,com} = o_m D_m F_m / f_n$, where f_n is the CPU cycle frequency of ENs in cluster n, which satisfies $f_n \in \{f_{PC}, f_{PV}, f_{MD}\}$. Thus, the partial computation energy consumption is $e^{n}_{m,com} = \kappa (f_n)^3 t^{n}_{m,com}$.

Accordingly, the total energy consumption for completing all tasks can be summed as

$$\begin{aligned} e_{total} = \sum_{m=1}^{M} \Bigg\{ & e^{local}_{m,com} + \left(\vartheta^{RSU}_m + \sum_{n=1}^{N} \vartheta^{n}_m \right) \left(P_v t^{up}_{m,V2R} + e^{RSU}_{m,com} + P_r t^{down}_{m,V2R} \right) \\ & + \sum_{n=1}^{N} \left[\vartheta^{n}_m \left(P_r t^{down}_{m,R2N_n} + e^{n}_{m,com} + P_e t^{up}_{m,R2N_n} \right) \right] \Bigg\}. \end{aligned} \tag{B2}$$

Obviously, the total energy consumption is related to task-splitting ratios, offloading strategies and the computation resource allocation scheme. With the above considerations, the optimization problem can be formulated to minimize the total energy consumption as follows:

$$\mathbf{P1}: \min_{\mathcal{R},\mathcal{S},\mathcal{F}} e_{total} \tag{B3a}$$

$$s.t.\quad l_m, o_m \in [0,1], \forall m \in \mathcal{M}, \tag{B3b}$$

$$l_m + o_m = 1, \tag{B3c}$$

$$\vartheta^{RSU}_m, \vartheta^{n}_m \in \{0,1\}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{B3d}$$

$$\vartheta^{RSU}_m + \sum_{n=1}^{N} \vartheta^{n}_m \leqslant 1, \sum_{m=1}^{M} \vartheta^{n}_m \leqslant K_n, \tag{B3e}$$

$$t^{local}_{m,com} \leqslant T_m, \tag{B3f}$$

$$t^{up}_{m,V2R} + t^{RSU}_{m,com} + t^{down}_{m,V2R} \leqslant T_{lim}, \tag{B3g}$$

$$t^{up}_{m,V2R} + t^{down}_{m,R2N_n} + t^{n}_{m,com} + t^{up}_{m,R2N_n} + t^{down}_{m,V2R} \leqslant T_{lim}, \tag{B3h}$$

$$\sum_{m=1}^{M} f_{m,RSU} \leqslant f_{RSU}, \tag{B3i}$$

$$f_{m,local} \leqslant f_{MD} \leqslant f_{PV} \leqslant f_{PC} < f_{RSU}, \tag{B3j}$$

Appendix B.2 The Game Approach for The Offloading Strategies

With the constraints of limited computation resources and tolerable delays, multi-vehicle offloading strategies interact with each other. Thus, it can be seen as a game, where all the TVs are the players. In this game, each player competes with the others for the computation resources to minimize energy consumption. The basic game of the offloading strategies can be defined as $\mathcal{G} = \{\mathcal{M}, \mathcal{S}, e_{total}\}$, where $\mathcal{M}$ is the player set of the game, $\mathcal{S}$ is the offloading strategies of players and e_{total} is the goal of the game, i.e. the utility function. For TV m, its offloading strategy is ϑ_m, hence, other TVs' offloading strategies can be denoted by $\vartheta_{-m} = \{\vartheta_1, ..., \vartheta_{m-1}, \vartheta_{m+1}, ..., \vartheta_M\}$. Thus, the utility function of TV m can be denoted by $e_{total}(\vartheta_m, \vartheta_{-m})$. The objective of each TV is to choose the most valuable offloading strategy to optimize its utility, that is, to minimize the total energy consumption, which can be formulated as

$$\min_{\{\vartheta_m | \forall m \in \mathcal{M}\}} e_{total}(\vartheta_m, \vartheta_{-m}). \tag{B4}$$

Definition 1 (The rule of change)**.** For a given $\mathcal{S}$, if TV m's offloading strategy ϑ_m is replaced by ϑ'_m and the result of the utility function satisfies $e_{total}(\vartheta'_m, \vartheta_{-m}) < e_{total}(\vartheta_m, \vartheta_{-m})$, this change is admitted.

Here, the concept of an NE is introduced in the game to derive the near-optimal offloading strategies.

Definition 2 (NE [12])**.** A matrix of offloading strategies $\mathcal{S}^*$ can be recognized to be the NE, if it is impossible for any TV $m \in \mathcal{M}$ to reduce the total energy consumption by changing its offloading strategy, i.e.

$$e_{total}(\vartheta^*_m, \vartheta^*_{-m}) \leqslant e_{total}(\vartheta_m, \vartheta^*_{-m}), \forall m \in \mathcal{M}. \tag{B5}$$

Definition 3 (Potential game [13])**.** If there is a potential function ϕ which satisfies (B6) for any TV $m \in \mathcal{M}$ and offloading strategy, the game can be called a potential game, which always has an NE.

$$\phi(\vartheta_m, \vartheta_{-m}) - \phi(\vartheta'_m, \vartheta_{-m}) = e_{total}(\vartheta_m, \vartheta_{-m}) - e_{total}(\vartheta'_m, \vartheta_{-m}). \tag{B6}$$

Corollary 1. The game approach for the offloading strategies is a potential game with a potential function and can converge the NE after finite iterations.

Proof. The corresponding potential function ϕ of the game for the offloading strategies can be expressed by

$$\begin{aligned}\phi\left(\mathcal{S}\right) =&\vartheta_m^{RSU}\sum_{\mathcal{M}}\left(e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}\right)\\&+\left(1-\vartheta_m^{RSU}\right)\left\{e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}+\sum_{n=1}^{N}\left[\vartheta_m^n\left(P_r t_{m,R2N_n}^{down}+e_{m,com}^{n}+P_e t_{m,R2N_n}^{up}\right)\right]\right.\\&\left.+\sum_{i=1,i\neq m}^{M}\left(e_{i,com}^{local}+P_v t_{i,V2R}^{up}+e_{i,com}^{RSU}+P_r t_{i,V2R}^{down}\right)\right\}.\end{aligned}\tag{B7}$$

If the partial task of TV m is calculated by the RSU or by the EN in cluster n, the values of the potential function are respectively given in (B8) and (B9),

$$\phi\left(\vartheta_m^{RSU},\vartheta_{-m}\right)=\sum_{\mathcal{M}}\left(e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}\right),\tag{B8}$$

$$\begin{aligned}\phi\left(\vartheta_m^{n},\vartheta_{-m}\right)=&e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}\\&+P_r t_{m,R2N_n}^{down}+e_{m,com}^{n}+P_e t_{m,R2N_n}^{up}+\sum_{i=1,i\neq m}^{M}\left(e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}\right).\end{aligned}\tag{B9}$$

According to (B8) and (B9), we can obtain

$$\begin{aligned}\phi\left(\vartheta_m^{RSU},\vartheta_{-m}\right)-\phi\left(\vartheta_m^{n},\vartheta_{-m}\right)=&\left(e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}\right)\\&-\left(e_{m,com}^{local}+P_v t_{m,V2R}^{up}+e_{m,com}^{RSU}+P_r t_{m,V2R}^{down}+P_r t_{m,R2N_n}^{down}+e_{m,com}^{n}+P_e t_{m,R2N_n}^{up}\right)\\&=e_{total}\left(\vartheta_m^{RSU},\vartheta_{-m}\right)-e_{total}\left(\vartheta_m^{n},\vartheta_{-m}\right).\end{aligned}\tag{B10}$$

Therefore, the game for the offloading strategies is an exact potential game, which has a pure-strategy NE at least.

Since the number of TVs and clusters is finite, and each TV can only choose at most another node to assist its task computation, the number of offloading strategies is finite. In addition, if the final offloading strategy matrix is not Nash-stable, there must be a condition that a TV can lower energy consumption by changing its offloading strategy, which is contradictory to **Definition 1** and the final strategies. Thus, the game can finally converge the NE after finite iterations.

Thus, **Corollary 1** is proved.

Appendix B.3 DLM-KKT for The Joint Optimization of Task-Splitting Ratios and Computation Resource Allocation

Based on a determined matrix of offloading strategies, the corresponding optimization problem of optimal task-splitting ratios and computation resource allocation is analyzed in this subsection.

If task Ω_m does not require computation offloading or the partial task is calculated on ENs, i.e. $\vartheta_m^{RSU}=0$ and $f_{m,RSU}=0$, according to (B2) and (B3j), making full utilization of TV m' local computation resources can result in less energy consumption. Consequently, the task-splitting ratio l_m satisfies

$$l_m=\begin{cases}\frac{f_{m,local}T_m}{D_m F_m}, & \frac{D_m F_m}{f_{m,local}}>T_m,\\ 1, & otherwise.\end{cases}\tag{B11}$$

It is assumed that the set of TVs whose offloaded partial tasks are calculated on RSU is $\mathcal{M}_r=\left\{m|\,\vartheta_m^{RSU}=1,m\in\mathcal{M}\right\}$ and the allocated computation resource from RSU are $\mathcal{F}_r=\{\,f_{m,RSU}\,|\,\forall m\in\mathcal{M}_r\}$. The energy consumption minimization problem of this set can be simplified as

$$\mathbf{P2}:\min_{\mathcal{R}_r,\mathcal{F}_r}\;e_{total}\tag{B12a}$$

$$s.t.\;\; l_m,o_m\in[0,1]\,,\forall m\in\mathcal{M}_r,\tag{B12b}$$

$$l_m+o_m=1,\tag{B12c}$$

$$t_{m,com}^{local}\leqslant T_m,\tag{B12d}$$

$$t_{m,V2R}^{up}+t_{m,com}^{RSU}+t_{m,V2R}^{down}\leqslant T_{lim},\tag{B12e}$$

$$\sum_{\mathcal{M}_r}f_{m,RSU}\leqslant f_{RSU},\tag{B12f}$$

where the total energy consumption of TVs in $\mathcal{M}_r$ can be expressed as

$$\begin{aligned}e_{total}\left(\mathcal{R}_r,\mathcal{F}_r\right)=&\sum_{\mathcal{M}_r}\left[\kappa(f_{m,local})^2 l_m D_m F_m+P_v t_{m,V2R}^{up}+\kappa(f_{m,RSU})^2 o_m D_m F_m+P_r t_{m,V2R}^{down}\right]\\\overset{(a)}{\approx}&\sum_{\mathcal{M}_r}\left[\kappa(f_{m,local})^2 l_m D_m F_m+P_v\frac{\left(1-l_m\right)D_m}{\overline{C}_{m,V2R}^{up}}+\kappa(f_{m,RSU})^2\left(1-l_m\right)D_m F_m+P_r\frac{\varsigma\left(1-l_m\right)D_m}{\overline{C}_{m,V2R}^{down}}\right],\end{aligned}\tag{B13}$$

where (a) utilizes the average rate of V2R uplink $\overline{C}_{m,V2R}^{up}=\int_0^{t_s}C_{m,V2R}^{up}\left(t\right)dt/t_s$ and downlink $\overline{C}_{m,V2R}^{down}=\int_0^{t_s}C_{m,V2R}^{down}\left(t\right)dt/t_s$ to simplify calculation, and here $t_s=2\sqrt{R^2-S^2}/v$.

Corollary 2. The above optimization problem **P2** is convex.

Proof. If **P**2 is a convex problem, the utility function shown as (B13) should be convex or concave, as well as the constraints shown as (B12b)∼(B12f). Obviously, the constraints (B12b)∼(B12d) and (B12f) are all linear functions, which satisfy the condition of the convex problem. For constraint (B12e) and utility function (B13), it is linear for l_m, hence it is just required to prove that they are convex or concave with respect to $f_{m,RSU}$.

Accordingly, the derivative of (B12e) with respect to $f_{m,RSU}$ can be derived by

$$\frac{\partial f}{\partial f_{m,RSU}} = -\frac{(1-l_m)\, D_m F_m}{(f_{m,RSU})^2}. \tag{B14}$$

The corresponding second-order derivative is

$$\frac{\partial^2 f}{\partial\, (f_{m,RSU})^2} = \frac{2\,(1-l_m)\, D_m F_m}{(f_{m,RSU})^3}, \tag{B15}$$

Due to $l_m \in [0,1]$, $\partial f^2/\partial\,(f_{m,RSU})^2 \geqslant 0$, thus constraint (B12e) is convex.

Here, to demonstrate the crenelation of (B13), it is differentiated with respect to $f_{m,RSU}$, shown as

$$\frac{\partial e_{total}}{\partial f_{m,RSU}} = 2\kappa f_{m,RSU}(1-l_m) D_m F_m. \tag{B16}$$

The second-order derivative can be calculated by

$$\frac{\partial^2 e_{total}}{\partial\,(f_{m,RSU})^2} = 2\kappa(1-l_m) D_m F_m. \tag{B17}$$

Due to $l_m \in [0,1]$, $\partial^2 e_{total}/\partial\,(f_{m,RSU})^2 \geqslant 0$. Thus, (B13) is a convex function.

Thus, **Corollary 2** is obtained.

For such a convex optimization problem, a Lagrange multiplier equation and its corresponding KKT constraints can be applied to obtain the optimal solution. Here, for constraint (B12b)∼(B12d), they can be incorporated as $l_m \in [0, l_{m,lim}]$, where $l_{m,lim} = \min\{T_m f_{m,local}/\,(D_m F_m)\,, 1\}$. The Lagrange multiplier equation can be written as

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{R}_r, \mathcal{F}_r, \beta_m, \gamma_m, \ell\right)\\
&= \sum_{\mathcal{M}_r}\left[\kappa (f_{m,local})^2 l_m D_m F_m + P_v \frac{(1-l_m)\, D_m}{\overline{C}^{up}_{m,V2R}} + \kappa (f_{m,RSU})^2\,(1-l_m)\, D_m F_m + P_r \frac{\varsigma\,(1-l_m)\, D_m}{\overline{C}^{down}_{m,V2R}}\right]\\
&\quad + \sum_{\mathcal{M}_r}\left\{\beta_m\left[l_m\left(l_m - l_{m,lim}\right)\right] + \gamma_m\left[\frac{(1-l_m)\, D_m}{\overline{C}^{up}_{m,V2R}} + \frac{(1-l_m)\, D_m F_m}{f_{m,RSU}} + \frac{\varsigma\,(1-l_m)\, D_m}{\overline{C}^{down}_{m,V2R}} - t_{m,lim}\right]\right\}\\
&\quad + \ell \cdot \left(\sum_{\mathcal{M}_r} f_{m,RSU} - f_{RSU}\right).
\end{aligned} \tag{B18}$$

Since there is a constraint on the total computation resources of the RSU, the complexity of the calculation is exponentially growing as the number of $\mathcal{M}_r$ if we utilize the traditional Lagrange multiplier equation. To reduce computational complexity, we assume that ℓ is a known value, and take advantage of dichotomy to reduce the deviation of its value and obtain the optimal $(\mathcal{R}_r, \mathcal{F}_r)$. For task Ω_m, its simultaneous equations for $(l_m, f_{m,RSU})$ based on KKT constraints can be shown as follows [14]:

$$\left\{\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial l_m} = 0, && \text{(B19a)}\\
&\frac{\partial \mathcal{L}}{\partial f_{m,RSU}} = 0, && \text{(B19b)}\\
&\beta_m\left[l_m\left(l_m - l_{m,lim}\right)\right] = 0, && \text{(B19c)}\\
&\gamma_m\left(t^{up}_{m,V2R} + t^{RSU}_{m,com} + t^{down}_{m,V2R} - t_{m,lim}\right) = 0, && \text{(B19d)}\\
&\ell\left(\sum_{i=1}^{m} f_{i,RSU} - f_{RSU}\right) = 0, && \text{(B19e)}\\
&\beta_m, \gamma_m \geqslant 0, && \text{(B19f)}
\end{aligned}\right.$$

where (B19a) and (B19b) are to calculate the extreme point, (B19c)∼(B19e) are the KKT-based complementary slackness of inequality constraints, and (B19f) is to limit Lagrange multipliers. According to the characteristic of KKT constraints, there are two conditions for an inequality constraint. Taking (B19c) as an example, one of the two conditions shown as follows must be satisfied:

$$\left\{\begin{aligned}
&l_m\left(l_m - l_{m,lim}\right) = 0, \beta_m > 0, && \text{(B20a)}\\
&l_m\left(l_m - l_{m,lim}\right) < 0, \beta_m = 0. && \text{(B20b)}
\end{aligned}\right.$$

It is the same for other inequality constraints.

To derive the solution for the joint optimization of task-splitting ratios and computation resource allocation, DLM-KKT is detailed in **Algorithm B1**.

Algorithm B1 DLM-KKT for The Joint Optimization of Task-Splitting Ratios and Computation Resource Allocation

Input: An initial offloading strategy matrix $\mathcal{S}_o$, task $\Omega_m = \{D_m, F_m, T_m\}$, $m \in \mathcal{M}$, computation resources of the RSU f_{RSU}, maximum tolerable deviation ϖ, and initial maximum and minimum ℓ_{max} and ℓ_{min}.

Output: The corresponding optimal task-splitting ratio matrix $\mathcal{R}^*$, computation resource allocation set $\mathcal{F}^*$ and energy consumption e^*_{total}.

```
while $\ell_{max} - \ell_{min} > \varpi$ do
    $\ell = \frac{\ell_{max}+\ell_{min}}{2}$;
    for $m = 1 : M$ do
        if $\vartheta_m^{RSU} = 0$ then
            Calculate $l_m^*$ according to (B11), and $f^*_{m,RSU} = 0$;
        else
            Construct the Lagrange multiplier equation and its corresponding KKT constraints based on the initial matrix $\mathcal{S}_o$ and $\Omega_m = \{D_m, F_m, T_m\}$ according to (B18) and (B19a)∼(B19f);
            Divide the constraints given by (B19c) and (B19d) into $2^2$ conditions as the example taken in (B20a) and (B20b);
            Build up the simultaneous equations;
            for $\mathcal{I} = 1 : 2^2$ do
                Solve the simultaneous equations expressed as (B19a)∼(B20b), whose solution vector is shown as $\left[l_m^{(\mathcal{I})}, f_{m,RSU}^{(\mathcal{I})}\right]$;
            end for
            Select the optimal solution $\left(l_m^*, f^*_{m,RSU}\right)$ that satisfies the constraints and minimizes the value of the utility function after being substituted into (B18);
        end if
    end for
    if $\sum_{i=1}^{m} f_{i,RSU} < f_{RSU}$ then
        $\ell_{max} = \ell$;
    else
        $\ell_{min} = \ell$;
    end if
end while
$\mathcal{R}^*$ and $\mathcal{F}^*$ are obtained;
Substitute $S_o$, $\mathcal{R}^*$ and $\mathcal{F}^*$ into (B2) to calculate $e^*_{total}$.
```

Appendix B.4 CORAJOA Based on Game Theory and DLM-KKT

Initially, a stochastic offloading strategy matrix is constructed as input. Subsequently, once all TVs take offloading strategies, the task-splitting ratios and computation resource allocation should be jointly optimized by DLM-KKT shown in **Alogorithm B1**. After the stage of joint optimization, the game approach updates the offloading strategies until it converges the NE, i.e. each TV has no desire to change its offloading strategy for energy consumption minimization.

In final, the near-optimal $\mathcal{R}$, $\mathcal{S}$, $\mathcal{F}$ and e_{total} can be derived by the CORAJOA after finite iterations shown in **Algorithm B2**. In particular, the complexity of DLM-KKT and CORAJOA can be calculated by $\mathcal{O}\left(2^2 \log_2\left(\ell_{max} - \ell_{min}/\varpi\right) * M\right)$ and $\mathcal{O}\left(\left(\mathcal{T}-1\right) M * \left(N+1\right) 2^2 \log_2\left(\ell_{max} - \ell_{min}/\varpi\right) * M\right)$ [15], respectively, where parameters are explained in **Algorithm B1** and **Algorithm B2**. From the complexity formulas, it is not difficult to obtain that the computational complexity will spike as maximum tolerable deviation ϖ decreases, though it favors computational accuracy.

Appendix C Simulation Results and Discussions

Initially, without special instructions, the task size is randomly selected from $D = [140, 160)$ (Kbits), the tolerable delay T_m is set to be $[100, 150]$ (ms), the total CPU cycle frequency of RSU f_{rsu} is 30 (GHz), and density λ_v is equal to 150 vehicles per kilometer. Other related parameters are presented in Table C1.

Table C1 Simulation Parameters

Parameter	Value	Parameter	Value
d_h (hard-core distance)	5 m	λ_c (density of clusters)	100 per km^2
R (coverage radius of RSU)	200 m	λ_p (density of ENs)	5
S (laid distance of RSU)	5 m	δ (fading parameter)	2
v (velocity of vehicles)	10 m/s	α_R, α_N (path fading exponent)	2, 4
P_v, P_r, P_n (transmit power)	30 dBm, 40 dBm, 30dBm	g_{ml}, g_{sl} (lobe gain)	18 dBi, -2 dBi
L_{mmw} (wavelength)	5 mm	θ (half of main lobe angle)	5°
B_{V2R}, B_{R2N} (bandwidth)	400 MHz, 400 MHz	ς (ratio of result and original task)	0.1
σ^2 (noise power)	-83 dBm	ϖ (maximum tolerable deviation)	0.001
F (computation density)	[2000,3000] cycle/bit	κ (energy consumption factor)	10^{-27}
$f_{m,local}, f_{MD}$ (cycle frequency)	[1.5,2] GHz, [2,2.5] GHz	f_{PV}, f_{PC} (cycle frequency)	[2.5,3] GHz, [3,3.5] GHz
ℓ_{max} (initial maximum of ℓ)	100	ℓ_{min} (initial minimum of ℓ)	0

To serve as a contrast, the following baseline schemes are considered. 1) Branch-and-bound algorithm (BBA) [16]: In this scheme, tasks are offloaded to the RSU or computed locally without consideration of ENs, and the computation resources of RSU are also allocated. The mixed integer nonlinear programming problem is transformed into a convex optimization problem for solution; 2) Mobility-aware computational efficiency based task offloading and resource allocation algorithm (MACTER) [17]: In [17], MACTER

Algorithm B2 CORAJOA Based on Game Theory and DLM-KKT

```
Input: The TV set $\mathcal{M}$, an initial offloading strategy matrix $\mathcal{S}_o$, task $\Omega_m = \{D_m, F_m, T_m\}$, $m \in \mathcal{M}$, total computation resources of the RSU $f_{RSU}$, the cluster set $\mathcal{N}$, and the number of ENs $\mathcal{K} = \{\mathcal{K}_1, ..., \mathcal{K}_n, ..., \mathcal{K}_N \,|\, \mathcal{K}_n = \{1, ..., K_n\}, \forall n \in \mathcal{N}\}$.
Output: The near-optimal task-splitting ratio matrix $\mathcal{R}$, computation resource allocation set $\mathcal{F}$, offloading strategy matrix $\mathcal{S}$, and energy consumption $e_{total}$.
Generate an offloading strategy matrix $\mathcal{S}(0)$ randomly;
Calculate the corresponding optimal $\mathcal{R}^*$, $\mathcal{F}^*$ and $e_{total}^*$ based on the initial $\mathcal{S}_o$ by Algorithm B1;
Set $\mathcal{T} = 1$, $\mathcal{S}(1) = \mathcal{S}_o$, $\mathcal{R}(1) = \mathcal{R}^*$, $\mathcal{F}(1) = \mathcal{F}^*$ and $e_{total}(1) = e_{total}^*$;
while $\mathcal{S}(\mathcal{T}) \neq \mathcal{S}(\mathcal{T}-1)$ do
    $\mathcal{T} = \mathcal{T} + 1$;
    $\mathcal{S}(\mathcal{T}) = \mathcal{S}(\mathcal{T}-1)$, $e_{total}(\mathcal{T}) = e_{total}(\mathcal{T}-1)$;
    for $m = 1 : M$ do
        for $n = 1 : N$ do
            $\vartheta_m^{RSU} = 0$, $\vartheta_m^n = 1$, and $\vartheta_m^j = 0, j \in \mathcal{N} \backslash n$;
            if $\sum_{i=1}^{M} \vartheta_i^n > K_n$ or Delay constraints cannot be satisfied then
                $e_{total}(\vartheta_m^n, \vartheta_{-m}) = \infty$;
            else
                $e_{total}(\vartheta_m^n, \vartheta_{-m})$ is calculated by (B2);
            end if
            if $\sum_{i=1}^{M} \vartheta_i^{RSU}$ is changed then
                Calculate the corresponding $\mathcal{R}^*$ and $\mathcal{F}^*$ by Algorithm B1;
            end if
            if $e_{total}(\vartheta_m^n, \vartheta_{-m}) < e_{total}(\mathcal{T})$ then
                $e_{total}(\mathcal{T}) = e_{total}(\vartheta_m^n, \vartheta_{-m})$;
                Update the offloading strategy change into $\mathcal{S}(\mathcal{T})$, as well as $\mathcal{R}^*$ into $\mathcal{R}(\mathcal{T})$ and $\mathcal{F}^*$ into $\mathcal{F}(\mathcal{T})$;
            end if
        end for
        Set $\vartheta_m^{RSU} = 1$ and $\vartheta_m^n = 0, \forall n \in \mathcal{N}$, and calculate the corresponding $\mathcal{R}^*$ and $\mathcal{F}^*$ by Algorithm B1;
        if $e_{total}\left(\vartheta_m^{RSU}, \vartheta_{-m}\right) < e_{total}(\mathcal{T})$ then
            $e_{total}(\mathcal{T}) = e_{total}\left(\vartheta_m^{RSU}, \vartheta_{-m}\right)$;
            Update the offloading strategy change into $\mathcal{S}(\mathcal{T})$, as well as $\mathcal{R}^*$ into $\mathcal{R}(\mathcal{T})$ and $\mathcal{F}^*$ into $\mathcal{F}(\mathcal{T})$;
        end if
    end for
end while
Derive $\mathcal{R}$, $\mathcal{F}$, $\mathcal{S}$ and $e_{total}$.
```

does not consider the ENs and utilizes the binary offloading. Here, partial offloading is introduced into this baseline method but the task-splitting ratios are not optimized; 3) Game-theoretic partial computation offloading algorithm (GT-PCO) [18]: This approach utilizes distributed computation offloading technology, however, it only considers ENs and the consideration of RSU is missing.

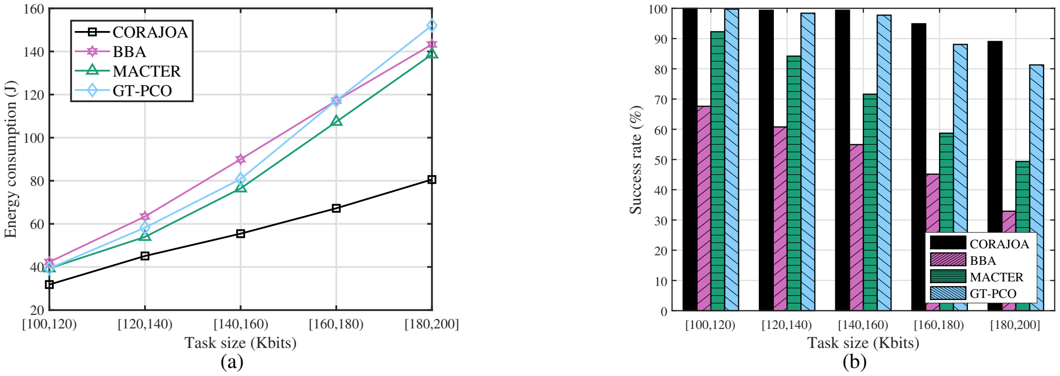

Figure C1 The performance versus task size under different schemes. (a) Energy consumption; (b) Success rate.

Figure C1 depicts the impact of the task size on the energy consumption and offloading success rate. From Fig. C1(a), it is obvious that the total system energy consumption is increasing as the task size enlarges. For all the schemes, the curves are nearly rising linearly while task size is growing evenly. This is because the expressions of transmitted energy consumption and computation energy consumption are both first-order functions with respect to task size. As illustrated in Fig. C1(b), the offloading success rate decreases as the task size enlarges, owing to the enhanced task data and ever-increasing required computation resources for delay constraints. According to Fig. C1, CORAJOA outperforms other schemes under arbitrary conditions that follow $D_m \in [100, 200]$ (Kbits) in both energy consumption and offloading success rate.

Figure C2 depicts the impact of the density of 1D PPP on the energy consumption and offloading success rate. It's worth noting that although λ_v is the density of initially generated dots, the actual vehicle density with the introduction of MHCPP is λ_h, which is calculated in (A1). As shown in Fig. C2(a), with the increasing of λ_v, the number of TVs is amplifying and the increasing trend

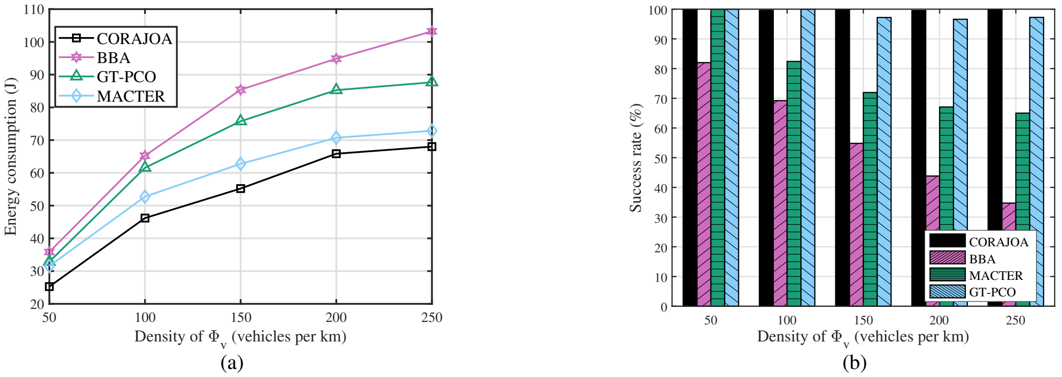

Figure C2 The performance versus density λ_v under different schemes. (a) Energy consumption; (b) Success rate.

is slowing down. In particular, the derivative of λ_h with respect to λ_v is $\partial\lambda_h/\partial\lambda_v = \exp(-2\lambda_v d_h)$ which is a monotone decreasing function, that is why the trend is slowing down. In addition, with the number of TVs enhancing, the offloading success rate is going down because of more intense competition for computation resources and more backlog of task data. In final, CORAJOA has the best performance among these schemes in energy consumption and offloading success rate under different λ_v.

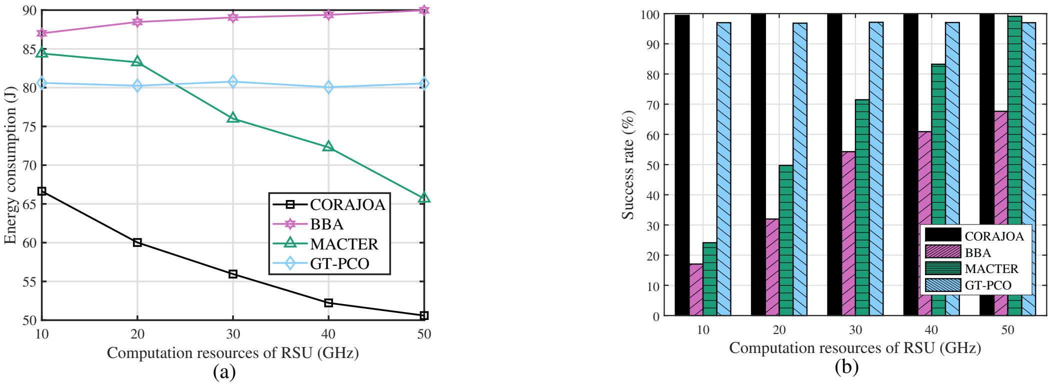

Figure C3 The performance versus total computation resources of the RSU under different schemes. (a) Energy consumption; (b) Success rate.

Figure C3 depicts the impact of the total computation resources of the RSU on the energy consumption and offloading success rate. Except for BBA and GT-PCO, the energy consumption decreases when the RSU is equipped with more computation resources, according to Fig. C3(a), that is because the task-splitting ratios and computation resource allocation can be optimized more rationally for energy minimization with richer computation resources benefiting from the distributed computation offloading technology. BBA does not consider partial offloading so it will cause more energy consumption when tasks are offloaded to RSU with the increasing computation resources for a higher offloading success rate. According to Fig. C3(b), it is not hard to determine that the richer computation resources can reach a higher offloading success rate. What is special is that the energy consumption and success rate of BBA do not change obviously since the RSU is not considered in this scheme. Besides, the proposed CORAJOA performs better than the other schemes under different f_{rsu}.

Figure C4 depicts the impact of the maximum tolerable time on the energy consumption and offloading success rate. As shown in Fig. C4(a), the total energy consumption diminishes as the tolerable delay enlarges, which is ascribed to the allocated computation resources becoming less when the delay constraints sustainably get looser. Accordingly, the computation energy consumption decreases. Besides, with the relaxation of the delay constraints and computation resources, more tasks can be calculated within the tolerable delay, achieving a higher offloading success rate. Furthermore, CORAJOA outmatches the other three schemes in both offloading success rate and energy cost under different delay constraints.

Compared with BBA, the results illustrate that distributed computing can lead to a significant reduction in energy consumption and an increase in offloading success rate. Different from MACTER, the system performance benefits from the optimization of task-splitting ratios and the introduction of ENs, even if it brings a much higher algorithmic complexity. Additionally, the comparison between MACTER and GT-PCO shows that ENs can increase the offloading success rate to a great extent, resulting in higher energy consumption, fortunately, CORAJOA can outperform better in both.

Figure C5 depicts the impact of the maximum tolerable deviation ϖ of the dichotomy on the energy consumption and offloading success rate. It is worth noting that the smaller ϖ leads to better performance both on the energy consumption and offloading success rate, which is attributed to the superior accuracy of the dichotomy. Nonetheless, the smaller ϖ also brings higher computational complexity, which is calculated in Appendix B.4, requiring better hardware performance urgently.

References

1 Yi W Q, Liu Y W, Nallanathan A. Coverage analysis for mmWave-enabled V2X networks via stochastic geometry. In: Proceedings of the IEEE Global Communications Conference, 2019. 1-6.

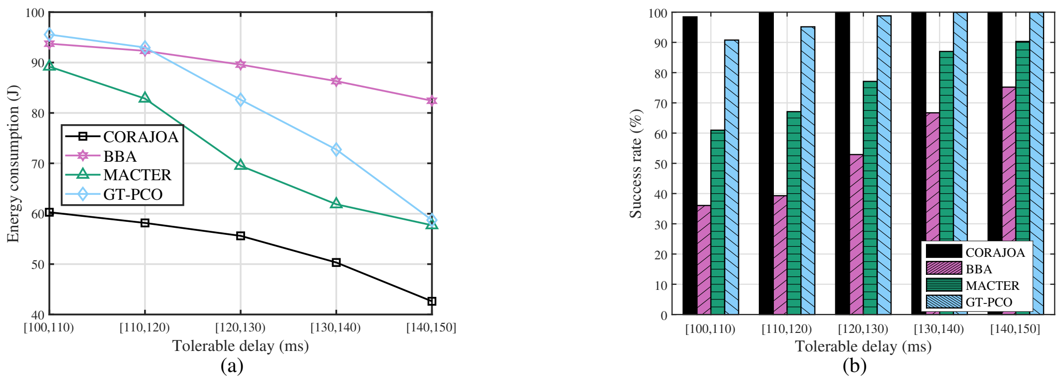

Figure C4 The performance versus tolerable time under different schemes. (a) Energy consumption; (b) Success rate.

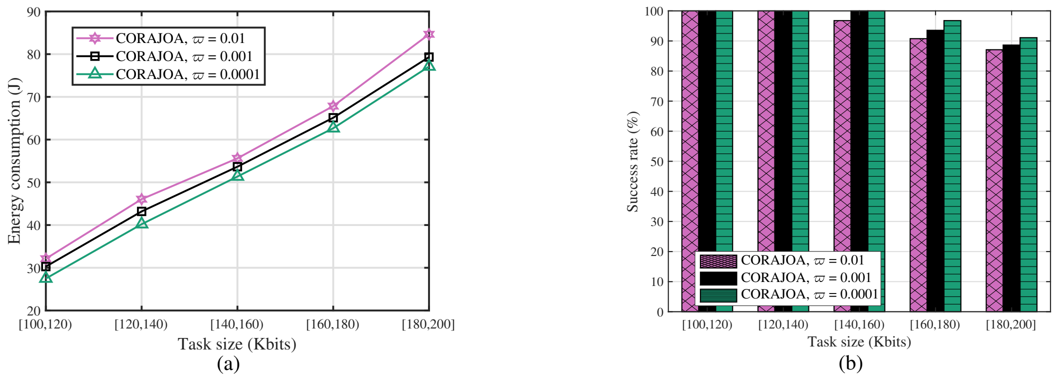

Figure C5 The performance versus task size under different ϖ. (a) Energy consumption; (b) Success rate.

2 Stoyan D, Kendall W S, Mecke J. Stochastic geometry and its applications. Hoboken, NJ, USA: Wiley, 1996.

3 Shi X M, Deng N, modelling and analysis of mmWave UAV swarm networks: A stochastic geometry approach. IEEE Trans Wireless Commun, 2022, 21(11): 9447-9459.

4 Wang Y B, Wu H, Niu Y, et al. Coalition game based full-duplex popular content distribution in mmWave vehicular networks. IEEE Trans Veh Technol, 2020, 69(11): 13836-13848.

5 Yi W Q, Liu Y W, Deng Y S, et al. Modeling and analysis of mmWave V2X networks with vehicular platoon systems. IEEE J Sel Areas Commun, 2019, 37(12): 2851-2866.

6 Fang Y, Pan Y C, Ma H, et al. A novel DCSK-based linear frequency modulation waveform design for joint radar and communication systems. IEEE Trans Green Commun Netw, 2024. doi: 10.1109/TGCN.2024.3422262.

7 Sun Y S, Ding Z G, Dai X C. On the outage performance of network NOMA (N-NOMA) modeled by Poisson line Cox point process. IEEE Trans Veh Technol, 2021, 70(8): 7936-7950.

8 Bafqi S F, Yazdi Z Z, Asadi A. Analytical Framework for Mmwave-Enabled V2X Caching. IEEE Trans Veh Technol, 2021, 70(1): 585-599.

9 Sun Y S, Ding Z G, Dai X C, et al. Performance of downlink NOMA in vehicular communication networks: An analysis based on Poisson line Cox point process. IEEE Trans Veh Technol, 2020, 69(11): 14001-14006.

10 Lin Z J, Chen X P, He X F, et al. Energy-efficient cooperative task offloading in NOMA-enabled vehicular fog computing. IEEE Trans Intell Transp Syst, 2024, 25(7): 7223-7236.

11 Mao Y Y, Zhang J, Letaief K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J Sel Areas Commun, 2016, 34(12): 3590-3605.

12 Cui J Y, Tang X J. A method for solving Nash equilibria of games based on public announcement logic. Sci China Inf Sci, 2010, 53: 1358–1368.

13 Zhang J, Xia W W, Yan F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing. IEEE Access, 2018, 6: 19324-19337.

14 Lin Z J, Fang Y, Chen P P, et al. Modeling and analysis of edge caching for 6G mmWave vehicular networks. IEEE Trans Intell Transp Syst, 2023, 24(7): 7422-7434.

15 Fang Y, Peng D Y, Ma H, et al. A neural network-aided detection scheme for index-modulation DCSK system. IEEE Trans Veh Technol, 2024, 73(2): 2109-2121.

16 Fan X G, Gu W T, Long C Q, et al. Optimizing task offloading and resource allocation in vehicular edge computing based on heterogeneous cellular networks. IEEE Trans Veh Technol, 2024, 73(5): 7175-7187.

17 Raza S, Wang S J, Ahmed M, et al. Task offloading and resource allocation for IoV using 5G NR-V2X communication. IEEE Internet Things J, 2022, 9(13): 10397-10410.

18 Pham X Q, The T H, Huh E N, et al. Partial computation offloading in parked vehicle-assisted multi-access edge computing: A game-theoretic approach. IEEE Trans Veh Technol, 2022, 71(9): 10220-10225.