Energy Minimization of Delay-Constrained Offloading in Vehicular Edge Computing Networks

Tianyu Yang, Yao Zhu, Yulin Hu*, and Rudolf Mathar
Institute for Theoretical Information Technology, RWTH Aachen University, D-52074 Aachen, Germany
Email: {yang, zhu, hu, mathar}@ti.rwth-aachen.de

***Abstract*—In this work, we study the energy efficiency problem in a vehicular edge computing network with delay sensitive tasks. We aim at minimizing the total energy consumption under the given delay constraints by jointly optimizing the offloading decisions and the computational resources. The resulting mixed integer non-convex problem is intractable to solve. To address the problem, it is reformulated into an equivalent tractable form and via relaxation it is transformed into a convex problem. Via simulations, the performance and the offloading behaviour of the proposed design are evaluated under various scenarios with different setups of vehicular density and delay limits.**

***Index Terms*—Vehicular edge computing (VEC), energy efficiency, delay aware.**

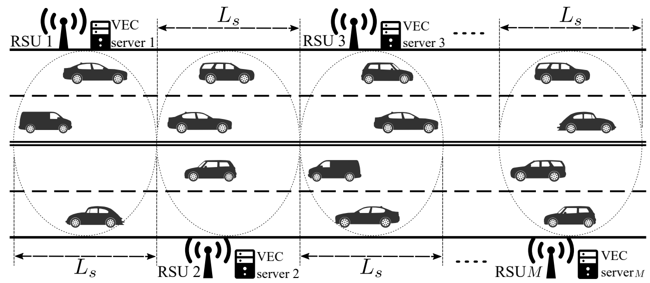

Figure 1. Illustration of the studied multi-lane bidirectional VEC network.

I. Introduction

Benefiting from the development of the Internet-of-things (IoT) and advanced vehicle-to-everything (V2X) communication technologies, vehicles can enjoy more intelligent and safe applications, e.g., autonomous driving and route planning. These advance service normally require high computational resources to process the computation-intensive and latency-critical tasks [1]. Nevertheless, the vehicles are resource constrained due to their power limit, and thus it is a challenge for them to achieve the desired quality of service and experience. In recent years, mobile edge computing (MEC) has emerged and fast developed. It allows the tasks with high computational and low latency requirements of the IoT devices to be offloaded to the edge computing nodes and be accomplished in the servers [2]–[6]. By deploying the vehicular edge computing (VEC) servers along with the road side units (RSUs), VEC networks can be established to overcome the computation constraints of the vehicles. Because of the proximity between the vehicles and VEC servers, the latency requirements can also be satisfied.

Many works have been proposed in recent years to cope with the offloading problem in VEC networks. In [7] a hierarchical VEC offloading framework is proposed, where the revenues of the VEC servers are maximized under given delay constraints of computing tasks. In [8] a similar framework is proposed to maximize the benefit of the MEC service provider while satisfying the delay requirements of the vehicles through a contract theoretic approach. These schemes are then improved in [9] to overcome the overload problem by balancing the load among VEC servers and minimizing the total system delay. Nevertheless, the aforementioned results are conducted under an assumption of static arrival of tasks, where all vehicles only generate request of tasks at the starting point of the VEC road, which is not practical in realtime networks. Moreover, unlike the cloud computing networks where the computational energy consumption of the cloud network can be ignored, in MEC networks, how to satisfy the computational and latency requirements with the consideration of energy efficiency is still a challenge [2]. More recently, a joint workload offloading and power control problem is formulated in [10], which minimizes the total energy consumption of the vehicles. However, the allocation strategy of the computational resources is still missing.

In this work, in contrast to the designs in [7]–[9], only considering the static VEC networks, or the designs in [10], without considering the computation resources optimization, we study a energy minimization problem in a dynamic VEC network by jointly optimizing the offloading decision and the computational resources.

II. System Model

We consider a multi-lane bidirectional VEC network, where M RSUs are successively and equidistantly located along the both road side, as shown in Fig.1. Each RSU is equipped with a VEC server. All RSUs are wired connected with a control center so that the received information of all RSU can be gathered and the offloading decision can be made globally. Each RSU has a wireless converge range and thus the whole VEC road is divided into M segments, where each has a length of L_s. Each VEC server is capable with a fix number

* Y. Hu is the corresponding author.

of V virtual machines (VMs), where each VM has a limited computational resource.

In order to deal with the realtime VEC networks, the offloading system operates under slotted time and has a updating time interval t_{sys}. At each updating time point the system receives N requests of the delay sensitive tasks from N corresponding vehicles and makes decision of the offloading strategy according to the status of the network, e.g., the location, speed and computation buffer of the vehicles, and the buffer of all VMs in the VEC servers. The n-th task, where $n \in \mathcal{N} = \{1, \ldots, N\}$ and $\mathcal{N}$ is the index set of all vehicles, can be represented as a tuple $(d_n, c_n, T_{\max,n})$, where d_n, c_n and $T_{\max,n}$ denote the data size, required frequency cycles to compute the task and the delay limit of the n-th task, respectively. Each task can be processed locally in the vehicle or offloaded to the RSU and accomplished in the VEC server. It is noted that if the task is offloaded to the server, the computation results should be transmitted back to the vehicle. Normally the data size of the computation results are very small compared to the offloaded data. Therefore, we do not consider the energy consumption and latency of the back-transmission process, as in [11]–[13]. However, it should be ensured that the vehicle is not out of the converge range of the chosen RSU during the back-transmission. This will be considered in section II-D.

Since the system is in realtime, the vehicles may make request at any position of the VEC road. Then, it is noted that the RSUs which behind the vehicle are not valid options for the offloading, since the vehicles only drive toward one direction. Therefore, the potential options for each vehicle are the RSUs ahead it along with the driving direction. On the other hand, if a vehicle is currently very close to the end of the VEC network, the available VEC servers for this vehicle is very limited, which incurs a high possibility of violation of the delay requirement. Therefore, for the fairness concern we only allow a fixed number Z of available VEC servers for each vehicle. Additionally, the vehicles, which do not have Z number of available VEC servers in the driving direction, are considered as out of the VEC network. Let $\mathcal{K} = \{1, \ldots, M \times V\}$ be the index set of all VMs. Then, we define $\mathcal{Z}_n \subset \mathcal{K}$ as the index set of all available VMs of the n-th vehicle. Note that the the set of all available VMs also includes the VM in the vehicle itself. Thus, the decision variable of the n-th vehicle's task to the k-th available VM, where $k \in \mathcal{Z}_n$ and $|\mathcal{Z}_n| = Z \times V + 1$, is denoted as $s_{n,k}$. Specifically, if the n-th vehicle decides to offload its task to the k-th available VM, then $s_{n,k} = 1$, otherwise $s_{n,k} = 0$. Specially, $s_{n,0}$ is the selection decision variable for the n-th vehicle itself, i.e., $s_{n,0} = 1$ means that the n-th vehicle computes its task locally. Note that for each vehicle the task should be computed at only one VM, i.e., the constraint $\sum_{k \in \mathcal{Z}_n} s_{n,k} = 1, \forall n \in \mathcal{N}$ should be satisfied.

A. Vehicular traffic and task arrival model

According to [14]–[16], the vehicular traffic model of a single lane follows a Poisson process with the parameter λ_l, where $l \in \mathcal{L}$ and $\mathcal{L}$ is the index set of all lanes. Additionally, the entire vehicle traffic model of the whole road also follows a Poisson process with the parameter $\lambda_s = \sum_{l \in \mathcal{L}} \lambda_l$ [17].

We assume that the offloading tasks of each vehicle follows the principle of the discrete time On/Off Markov arrival model [18], which is represented as a probability p at each time instance.

B. Vehicle computation model

If it is decided that the n-th vehicle computes its task locally, the selection decision variable $s_{n,0} = 1$ and a computational resource $f_{n,0}$ is allocated for the task. Thus, the computation delay of the task is $t^{\text{c}}_{n,0} = c_n / f_{n,0}$. We consider that the computation buffer of the vehicle may has the task of last time instant, and thus the task of this time instant can only be computed after the accomplishment of last task. This leads to a waiting delay $t^{\text{w}}_{n,0}$. Then, the total delay of vehicle computation process is

$$\begin{aligned} t_{n,0} &= t^{\text{w}}_{n,0} + t^{\text{c}}_{n,0} \\ &= t^{\text{w}}_{n,0} + \frac{c_n}{f_{n,0}}. \end{aligned} \tag{1}$$

C. VEC computation model

When the n-th vehicle chooses the k-th valid VM from its available VM set of the server, i.e., $k \in \mathcal{Z}_n \setminus \{0\}$, to offload its task, the selection decision variable $s_{n,k} = 1$, and the offloading process consists of 4 different delays as explained in the follows.

1) Driving delay: Firstly, the vehicles should drive into the coverage range of the chosen RSU. This leads to a driving delay

$$t^{\text{v}}_{n,k} = \frac{L_{n,k}}{v_n}, \tag{2}$$

where $L_{n,k}$ is the distance between the current position of the vehicle and the boundary of the chosen RSU's coverage range, v_n is the speed of n-th vehicle. Note that if the vehicle is currently located inside the coverage range of the chosen RSU, $L_{n,k} = 0$ and thus $t^{\text{v}}_{n,k} = 0$.

2) Transmission delay: Secondly, the vehicles should transmit the task data to the chosen VEC server. This leads to a transmission delay

$$\begin{aligned} t^{\text{r}}_{n,k} &= \frac{d_n}{r_{n,k}} \\ &= \frac{d_n}{B \log_2\left(1 + \frac{P_n h_{n,k}}{N_0}\right)}, \end{aligned} \tag{3}$$

where $r_{n,k}$ is the transmission rate, B is the bandwidth, P_n is the transmission power of the n-th vehicle, N_0 is the noise power and $h_{n,k}$ is the channel gain between the n-th vehicle and the chosen RSU. We consider the Line-of-Sight (LoS) channel model and thus the channel gain can be predicted with the known distance.

3) Waiting delay: Similarly as the computation process in the vehicles, we consider the waiting delay in the computation buffer of the VMs. Note that during the driving time and the transmission time the computation process of the task from last time instant in all VMs is still ongoing. Therefore, the waiting delay should exclude the driving and transmission time, which can be obtained as

$$t^{\mathrm{w}}_{n,k} = \max\left\{t^{\mathrm{b}}_{n,k} - t^{\mathrm{v}}_{n,k} - t^{\mathrm{r}}_{n,k},\ 0\right\}, \tag{4}$$

where $t^{\mathrm{b}}_{n,k}$ is the remaining computation time of the task from last time instant at the decision time point.

4) Computation delay: Finally, the task is computed in the VM under the allocated computation resource $f_{n,k}$. Thus, the computation delay is obtained as

$$t^{\mathrm{c}}_{n,k} = \frac{c_n}{f_{n,k}}. \tag{5}$$

D. Total time delay with selection decision variables

For the convenience of the formulation, we introduce the constants $t^{\mathrm{v}}_{n,0} = 0$ and $t^{\mathrm{r}}_{n,0} = 0$ to denote the zero driving and transmission delay in the local vehicle computation model. Then, the total time delay of the n-th task in k-th VM ($k \in \mathcal{Z}_n$) is obtained as

$$t_{n,k} = t^{\mathrm{v}}_{n,k} + t^{\mathrm{r}}_{n,k} + t^{\mathrm{w}}_{n,k} + t^{\mathrm{c}}_{n,k}. \tag{6}$$

With the consideration of the selection decision variables, the total time delay of the n-th vehicle's task is expressed as

$$T_n = \sum_{k\in\mathcal{Z}_n} s_{n,k} t_{n,k}. \tag{7}$$

As mentioned before, it should be guaranteed that the vehicles do not move out of the coverage range of the chosen RSUs to process the back-transmission. Thus, the true delay limit of the n-th task related to the k-th VM ($k \in \mathcal{Z}_n \setminus \{0\}$) is obtained as $\tilde{T}_{\mathrm{max},n,k} = \min\{T_{\mathrm{max},n}, T_{\mathrm{out},n,k}\}$, where $T_{\mathrm{out},n,k}$ is the time duration for the n-th vehicle driving out of the coverage range of the k-th VM. Since there is no back-transmission process for local computation, we have $\tilde{T}_{\mathrm{max},n,0} = T_{\mathrm{max},n}$.

E. Energy consumption model

We consider two types of the energy consumption in our VEC network: the computational energy consumption and the transmission energy consumption. Specifically, if the task is computed locally in the vehicle, a computational energy consumption is costed related to the workload of the task and the allocated computational resource of the vehicle. Oppositely, if the task is decided to be offloaded to the server, the transmission energy consumption of the vehicle and the computational energy consumption of the server are taken into account.

For the n-th vehicle, the energy consumption to transmit the task to the k-th VM is expressed as

$$E^{\mathrm{r}}_{n,k} = t^{\mathrm{r}}_{n,k} P_n. \tag{8}$$

On the other hand, the computational energy consumption for the n-th task with the total required computational cycles c_n and a given computational frequency $f_{n,k}$ is expressed as [19]

$$E^{\mathrm{c}}_{n,k} = \kappa c_n f^2_{n,k}, \tag{9}$$

where κ is a hardware related parameter.

III. Joint Computation Resource and Offloading Decision Optimization

In this section we aim to minimize the total energy consumption with the satisfaction of the delay constraints by jointly optimizing the offloading strategy and the computational resources. In practice, the energy consumption of the VEC servers may be less important compared to the energy consumption of the vehicles. Therefore, without loss of generality we add a weighting factor $\alpha_k \in [0,1]$ to the consumed energy of each VM, where $\alpha_k = 0$ represents that the energy consumption of the k-th VM is not concerned. Then, the resulting optimization problem is formulated as

$$\begin{aligned}
\min_{\mathbb{S},\mathbb{F}} \quad & \sum_{n\in\mathcal{N}}\sum_{k\in\mathcal{Z}_n} s_{n,k}\left(\alpha_k E^{\mathrm{c}}_{n,k} + E^{\mathrm{r}}_{n,k}\right) && \text{(10a)}\\
\text{s.t.} \quad & T_n \le \sum_{k\in\mathcal{Z}_n} s_{n,k}\tilde{T}_{\mathrm{max},n,k}, \quad \forall n\in\mathcal{N}, && \text{(10b)}\\
& 0 \le f_{n,k} \le s_{n,k} F_{\mathrm{max},k}, \quad \forall n\in\mathcal{N},\ k\in\mathcal{Z}_n, && \text{(10c)}\\
& \sum_{k\in\mathcal{Z}_n} s_{n,k} = 1, \quad \forall n\in\mathcal{N}, && \text{(10d)}\\
& \sum_{n\in\mathcal{N}} s_{n,k} \le 1, \quad \forall k\in\tilde{\mathcal{Z}}, && \text{(10e)}\\
& s_{n,k}\in\{0,1\}, \quad \forall n\in\mathcal{N},\ k\in\mathcal{Z}_n, && \text{(10f)}
\end{aligned}$$

where $\mathbb{S}$ and $\mathbb{F}$ are the set of all selection decision variables and computational resources, respectively. The constraint (10b) ensures that the time delay of each task should not exceed the corresponding delay limit. (10c) gives the constraints of the computational resources. Note that in (10c) if $s_{n,k} = 0$, then $f_{n,k} = 0$. This ensures that for the VMs without assigned tasks they are not allocated with any computational resource. Furthermore, (10d) guarantees that for each task only one VM of the server or the vehicle can be chosen. In (10e), $\tilde{\mathcal{Z}} = \bigcup_{n\in\mathcal{N}} \mathcal{Z}_n$ represents the union set of all index sets of available VMs. (10e) guarantees that for each VM only one task can be assigned.

It is noted that the optimization problem in (10) contains some intractable terms that are not jointly convex over $s_{n,k}$ and $f_{n,k}$, i.e., the terms $s_{n,k}E^{\mathrm{c}}_{n,k} = \kappa c_n s_{n,k} f^2_{n,k}$ in the objective function (10a) and the terms $s_{n,k}t^{\mathrm{c}}_{n,k} = s_{n,k}\frac{c_n}{f_{n,k}}$ in the constraint (10b). Moreover, due to the binary constraints in (10f) the optimization problem is a mixed integer non-convex problem. To solve the problem, we firstly reformulate it as the following equivalent form and then relax the binary variables to real variables, finally we proof that the resulting relaxed problem is a joint convex problem.

$$\min_{\mathbb{S},\mathbb{F}} \quad \sum_{n\in\mathcal{N}}\sum_{k\in\mathcal{Z}_n}\left(\alpha_k E^{\mathrm{c}}_{n,k} + s_{n,k}E^{\mathrm{r}}_{n,k}\right) \tag{11a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{Z}_n} s^2_{n,k}t_{n,k} \leq \sum_{k\in\mathcal{Z}_n} s_{n,k}\tilde{T}_{\max,n,k},\ \forall n\in\mathcal{N}, \tag{11b}$$

$$\text{(10c), (10d), (10e), (10f).}$$

Lemma 1. *The optimization problems in (10) and (11) are equivalent.*

Proof. Since $s_{n,k}$ is a binary variable we evaluate two situations. When $s_{n,k}=0$, $s^2_{n,k}t_{n,k}=s_{n,k}t_{n,k}=0$, (10c) implicates that $f_{n,k}=0$ and thus $s_{n,k}E^{\mathrm{c}}_{n,k}=E^{\mathrm{c}}_{n,k}=0$. Oppositely, when $s_{n,k}=1$, $s^2_{n,k}t_{n,k}=s_{n,k}t_{n,k}=t_{n,k}$ and $s_{n,k}E^{\mathrm{c}}_{n,k}=E^{\mathrm{c}}_{n,k}$. For any possible value of $s_{n,k}$ we have $s^2_{n,k}t_{n,k}=s_{n,k}t_{n,k}$ and $s_{n,k}E^{\mathrm{c}}_{n,k}=E^{\mathrm{c}}_{n,k}$. Thus (10) and (11) are equivalent. □

Then we relax the binary variables $s_{n,k}$ to real variables with the range of $[0,1]$ and get the following optimization problem.

$$\begin{aligned}\min_{\mathbb{S},\mathbb{F}} \quad & \sum_{n\in\mathcal{N}}\sum_{k\in\mathcal{Z}_n}\left(\alpha_k E^{\mathrm{c}}_{n,k} + s_{n,k}E^{\mathrm{r}}_{n,k}\right)\\ \text{s.t.} \quad & \text{(11b), (10c), (10d), (10e),}\\ & 0\leq s_{n,k}\leq 1, \quad \forall n\in\mathcal{N},\ k\in\mathcal{Z}_n.\end{aligned} \tag{12}$$

Lemma 2. *The relaxed optimization problem in (12) is a joint convex problem.*

Proof. Note that the objective function of (12) is a convex function over $\mathbb{S}$ and $\mathbb{F}$. The constraints (10c), (10d) and (10e) are also linear convex constraints. In (11b) the non-linear term $s^2_{n,k}t^{\mathrm{c}}_{n,k}=s^2_{n,k}\frac{c_n}{f_{n,k}}$ consists with both $s_{n,k}$ and $f_{n,k}$. Let $g\left(s_{n,k},f_{n,k}\right)=\frac{s^2_{n,k}}{f_{n,k}}$, then the Hessian matrix of $g\left(s_{n,k},f_{n,k}\right)$ is $\mathbf{H}=\begin{bmatrix}h_{11} & h_{12}\\ h_{21} & h_{22}\end{bmatrix}=\begin{bmatrix}\frac{2}{f_{n,k}} & -\frac{2s_{n,k}}{f^2_{n,k}}\\ -\frac{2s_{n,k}}{f^2_{n,k}} & \frac{2s^2_{n,k}}{f^3_{n,k}}\end{bmatrix}$. With the conditions $s_{n,k}\geq 0$ and $f_{n,k}>0$, we have $h_{11}>0$, $h_{22}\geq 0$ and $|\mathbf{H}|=h_{11}h_{22}-h_{12}h_{21}=\frac{4s^2_{n,k}}{f^4_{n,k}}-\frac{4s^2_{n,k}}{f^4_{n,k}}=0$. Thus $g\left(s_{n,k},f_{n,k}\right)$ is a joint convex function over $s_{n,k}$ and $f_{n,k}$. Combining with the other convex terms, (12) is a joint convex problem. □

Since the relaxed problem is a convex problem, the original problem in (11) is solvable via the advanced solvers that can solve mixed integer convex problem. Because of the constraint in (10d) the computational complexity can be reduced to $\mathcal{O}\left(N(Z\times V+1)\right)$.

IV. Simulation Results

In this section, the proposed VEC network design is numerically evaluated. We consider a 4 lane bidirectional road with $M=8$ RSUs located along the both sides. The lane width is 4m. The coverage range of each RSU is $L_s=30$m. Each VEC server is equipped with $V=4$ VMs. The number of available VEC servers for each vehicle is $Z=4$. The computational limit $F_{\max}$ of each VM in the VEC servers is 5GHz and in the vehicles is 1.2GHz. The speed of the vehicles is assumed to be constant as $v=120$km/h. Moreover, the task size, required computational resources and the delay limit of each task follow the uniform distributions $U(10,30)$MB, $U(1,5)$GHz and $U(8,10)$s, respectively. For the communication channel model, we adopt the LoS path loss model in [20] as $\mathrm{PL}=32.45+20\log_{10}f_0(\mathrm{MHz})+20\log_{10}d(\mathrm{km})$, where f_0 is the carrier frequency in MHz and d is the distance in km. Furthermore, we assume that the bandwidth of each transmission is $B=1$MHz, the carrier frequency is $f_0=5$GHz, the transmit power of each vehicle is $P=20$dBm and the noise power is -174dBm/Hz. Following [19] we set $\kappa=10^{-11}$ for the computational energy consumption. We fully concern all consumed energy equally as $\alpha_k=1$, $\forall k\in\mathcal{K}\cup\{0\}$. The updating interval is $t_{\mathrm{sys}}=2$s and the resulting system performance is averaged over 100 updating times, i.e., 200 seconds.

We firstly evaluate the consumed energy for processing the tasks of the whole VEC network. In Fig. 2 the average energy consumption of each updating time is depicted for the cases with various vehicular density and task arrival probability p. It is noticed that the average energy consumption of the whole VEC network is increasing as the vehicular density increases. However, for different task arrival level the increment speed of the consumed energy is different. Specifically, under low task rate the consumed energy increases slowly with the increment of the vehicular density. Oppositely, under high task rate the consumed energy increases dramatically. For instance, while the vehicular density changes from about 20 veh/km to 190 veh/km, under low task rate, i.e., $p=0.2$, the consumed energy increases from about 0.012J to 0.124J, under high task rate, i.e., $p=0.8$, the consumed energy increases from about 0.137J to 1.29J. This results in that the corresponding difference between the consumed energy under low task rate and high task rate grows from 0.1257J at 20 veh/km to 1.166J at 190 veh/km. This indicates that the task arrival rate influences the consumed energy significantly.

In Fig. 3 and Fig. 4 the impacts of the delay limit $T_{\max}$ on the energy consumption and offloading ratio are depicted, respectively. We set the vehicular density to 100 veh/km and the delay limit for all task in a range of [3,10] seconds. It is observed from Fig. 3 that as the delay limit decreases the consumed energy increases dramatically for all task arrival levels under low delay limit range, i.e., 3-5 second. This is since that lower delay limit requires higher computational resources, which leads to a larger energy consumption for computation. Nevertheless, when delay limit is larger than 7 second, the consumed energy does not changes so much. This is because that a longer computation time in the VMs leads to a longer waiting delay for next time instant, and thus the task in next time instant requires a shorter computation time to satisfied the same delay limit level, which requires a

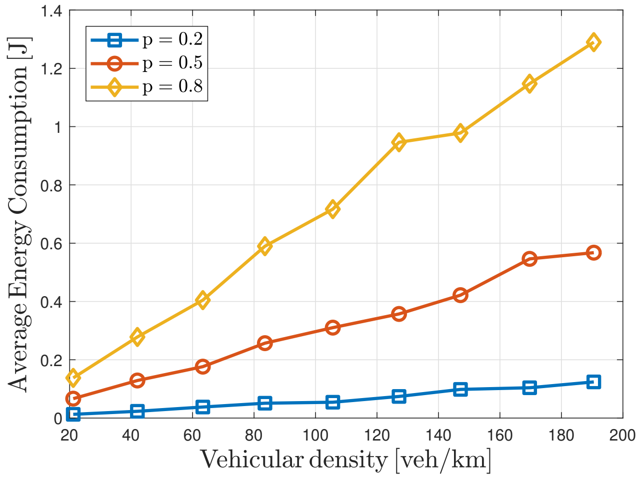

Figure 2. Average energy consumption vs. vehicular density.

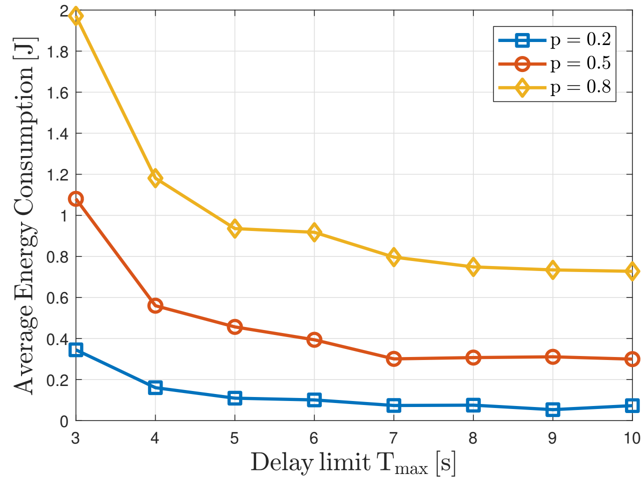

Figure 3. Average energy consumption vs. delay limit $T_{\max}$.

larger computational resources, i.e., larger energy consumption. Moreover, it is noticed that a higher task rate leads to a more significant change of the consumed energy through the whole delay limit range.

From Fig. 4 we evaluate the offloading behaviour of the proposed VEC network under various delay limits. Note that the depicted offloading ratio is a averaged behaviour over many time instance. For a single time instance with a dense task flow the offloading ratio can reach a relatively high level, e.g., 0.8. It is observed that generally the offloading trend increases as the delay limit decreases. Under very low delay limits, e.g., $T_{\max} = 3$s, the offloading ratio increases dramatically. This is because the computational capability of the vehicles is much weaker than the computational capability of the VEC servers, thus the vehicles can not achieve the delay requirements locally when the delay limits are very short. Moreover, similarly as the results related to the energy consumption, the increasing of the task arrival rate lead to a significant increment of offloading ratios.

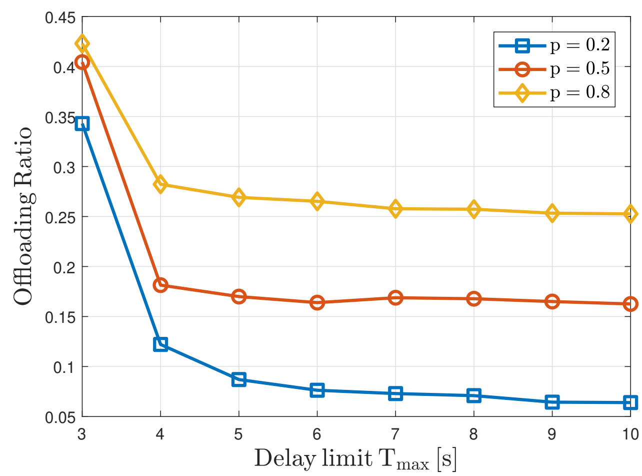

Figure 4. Offloading ratio vs. delay limit $T_{\max}$.

V. Conclusion

In this paper, we studied a energy efficiency problem in the VEC network with delay constrained tasks. By considering the computation and transmission energy consumption we aimed to minimize the total consumed energy under the delay constraints via the joint optimization of the offloading strategy and computational resources allocation. To cope the resulting mixed integer non-convex problem, we provided a relaxed problem and proof its convexity. Via the numerical evaluation we analyzed the behaviour of the energy consumption and offloading under various of the vehicular densities and delay limits.

References

[1] C. Wang, Y. Li, D. Jin, and S. Chen, "On the serviceability of mobile vehicular cloudlets in a large-scale urban environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2960–2970, 2016.

[2] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[3] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Transactions on Vehicular Technology*, 2019.

[4] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 93–107, 2018.

[5] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[6] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, "BEGIN: Big data enabled energy-efficient vehicular edge computing," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 82–89, 2018.

[7] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[8] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*. IEEE, 2016, pp. 288–294.

[9] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint offloading and resource allocation in vehicular edge computing and networks," *arXiv preprint arXiv:1807.08718*, 2018.

[10] Z. Zhou, P. Liu, Z. Chang, C. Xu, and Y. Zhang, "Energy-efficient workload offloading and power control in vehicular edge computing," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2018 IEEE*. IEEE, 2018, pp. 191–196.

[11] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, no. 5, pp. 2795–2808, 2016.

[12] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[13] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE access*, vol. 4, pp. 5896–5907, 2016.

[14] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3341–3356, 2008.

[15] M. Khabazian and M. K. M. Ali, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2440–2450, 2008.

[16] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, "A simple free-flow traffic model for vehicular intermittently connected networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1312–1326, 2012.

[17] E. Cascetta, *Transportation systems engineering: theory and methods*. Springer Science & Business Media, 2013, vol. 49.

[18] M. Ozmen and M. C. Gursoy, "Wireless throughput and energy efficiency with random arrivals and statistical queuing constraints," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1375–1395, 2016.

[19] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[20] G. Charalampopoulos, T. Dagiuklas, and T. Chrysikos, "V2i applications in highways: How rsu dimensioning can improve service delivery," in *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE, 2016, pp. 1–6.