2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 978-1-6654-0527-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/CSCWD54268.2022.9776123

An Extended Adaptive Large Neighbourhood Search for Vehicles' Task Offloading in Platooning

Hongna Lou
School of Mechanical Engineering
Dalian University of Technology
Dalian, China
hongna.lou@foxmail.com

Fangyi Hu
Avic Research Institute For Special Structures Of Aeronautical Composites
Jinan, China
hufy_2021@163.com

Jiajian Li
School of Mechanical Engineering
Dalian University of Technology
Dalian, China
lijiajian@mail.dlut.edu.cn

Xiaojun Zheng
School of Mechanical Engineering
Dalian Jiaotong University
Dalian, China
cnzhengxj@163.com

Yanjun Shi
School of Mechanical Engineering
Dalian University of Technology
Dalian, China
syj@ieee.org

***Abstract*—With the arrival of the Internet of Things, many fragmented intelligent terminals unload data to the edge cloud for computing. Based on the background of the vehicle platooning assisted by the edge cloud server, this paper conducted the following research on the problem of computing task offloading: First, considering the limited heterogeneous network resources of the Internet of Vehicles, the limited computing resources allocated by the edge cloud server for the vehicle platoon and member vehicle onboard computing unit, and the different delay constraints of the computing tasks, a computation offloading model was established with the optimization objective of reducing the total energy consumption of the platooning. Second, this paper used the proposed extended adaptive large neighbourhood search (EALNS) algorithm to optimize the offloading decision of computing tasks and used a method that enables computing tasks to be completed within the time delay constraint to optimize the allocation of computing resources. Finally, the EALNS and the generalized Benders decomposition algorithm were compared for energy optimization experiments. The experimental results verified the EALNS algorithm's effectiveness in optimizing the platooning's total energy consumption in the task offloading decision-making process.**

***Keywords*—connected and automated vehicles; cooperative vehicle infrastructure; Computation Offloading; Extended Adaptive Large Neighbourhood Search**

I. Introduction

With the rapid development of mobile communication technology and the widespread application of the internet of things (IoT), the era of IoT is bringing a whole new world. At the same time, the vehicle is undergoing critical changes, in which many smart mobile services are used, such as augmented reality[1], computer vision, and so on. These services require intensive communication and computation resources[2-4]. Cloud computing with high-powered computing capability has served as an effective approach to solving the conflict between resource-intensive applications and resource-constrained vehicles, where the tasks can be offloaded to the remote cloud servers from a vehicle[5-7]. However, the frequent interaction process between the vehicles and the remote cloud will increase the burden on the network, resulting in the blockage of the critical path of the core network, which cannot meet the increasing demand for low latency of the services. For this reason, mobile edge computing (MEC) is proposed as a promising solution, which pushes remote cloud services to the edge of the wireless network. Placing computing resources as close as possible to the terminal realizes context awareness and real-time computing capabilities.

In recent years, research on edge computing has mainly focused on offloading technology, resource management technology, mobility management, and security and privacy protection technology. There is no clear boundary between edge computing offloading technology and other technologies. This technology particularly focuses on improving system performance, such as reducing energy consumption or computing task completion time by optimizing offloading decisions and effectively allocating resources. Unlike the remote cloud, the edge cloud is limited by the external environment of the base station, and its own deployment cost, so distributed small-scale servers implement it. More fragmented smart terminals will be connected to the edge cloud for computing. The performance of traditional network methods on the Internet of Vehicles (IoV) will suddenly drop. Therefore, under the constraints of limited computing resources, limited network resources, and onboard application delays, reasonably optimizing the offloading of computing tasks and allocating computing resources to reduce the total energy consumption of

This work was supported by China National Key Research and Development Program (NO.2018YFE0197700).

vehicles has become an important issue. This paper uses the vehicle platooning assisted by the MEC server as the research background to study computation offloading in the cooperative vehicle infrastructure system.

This article takes the vehicle platooning under the cooperative vehicle infrastructure system as the scene. The main research contents of this paper are as follows: First, this paper considers the limited heterogeneous network resources of IoV, the limited computing resources allocated by the MEC server to the vehicle formation, the limited computing resources of the onboard units (OBU) of member vehicles, and the different delay constraints of computing tasks. The offloading problem of cooperative computing tasks is modelled as an integer nonlinear programming model. Secondly, an extended adaptive large neighbourhood search algorithm (EALNS) is used for the computing task offloading decision. A method that enables the computing task to be completed within the time delay constraint is adopted for the computing resource allocation strategy. The paper minimizes the total energy consumption of the platooning by optimizing the offloading of computing tasks and allocating computing resources. Finally, the simulation experiment of vehicle-infrastructure collaborative computing task offloading based on EdgeCloudSim verifies the algorithm's effectiveness.

The remainder of this paper is structured as follows. Related work is introduced in Section 2. Model analysis of task offloading based on cooperative vehicle infrastructure system is described in Section 3. Computing task offloading decision based on extended adaptive large neighbourhood search (EALNS) is shown in Section 4. Section 5 presents the simulation results. The conclusion is mentioned in Section 6.

II. Related Work

As a key technology and a new network computing paradigm to help 5G, edge computing has always been a hot issue of widespread concern. According to the research literature related to the computing offloading of IoV, the common edge cloud computing resources in IoV include (a) computing resources available to nearby vehicles, also called in-vehicle edge cloud computing resources; (b) computing resources available to the RSU server; (c) The computing resources available to the MEC server on the base station side.

For in-vehicle edge cloud computing resources, it integrates the computing resources of vehicles and provides computing services. Sun et al.[8] proposed an adaptive learning task offloading algorithm based on a multi-armed gambling machine. The candidate vehicles for each time slice are selected based on the historical time delay observation data to minimize the total completion time of the computing task in a period.

For the computing resources of the roadside unit (RSU) server, due to the limitation of the communication distance between the vehicle and the RSU, it is necessary to consider switching the RSU connected to the vehicle. The calculation task may not be offloaded to the RSU server where the vehicle is located. Xu et al.[9] used the computing task offloading plan (the RSU server selection for each vehicle computing task) as a decision variable to minimize the total time loss and the energy consumption of all servers, and solved multi-objective constrained optimization problems by improving the NSGA-II algorithm. Yang et al.[10] considered that each vehicle has an in-vehicle application program containing many subtasks to execute, and proposed a location-based offload scheme. By considering the delay of each computing task and the constraints of RSU server computing resources, Zhang et al.[11] proposed a contract-based offloading server selection and computing resource allocation scheme to maximize the benefits of suppliers and improve the utility of vehicles. Dai et al.[12] aimed to minimize the task processing delay (including the vehicle's movement time from the starting point to the target RSU) and balance RSU server load. The mixed-integer nonlinear programming problem was transformed into selecting the offloading server for each vehicle computing task, determining the offloading ratio, and allocating server computing resources.

The computing resources of the MEC server are usually divided into a single MEC server and multiple MEC servers of neighbouring base stations for collaborative computing. To optimize energy efficiency and service quality, Zhang et al.[13] proposed a regionally coordinated IoV architecture based on Fog Computing. Four services were designed: mobility support and task migration, multi-source data collection, distributed computing and storage, and multi-path data transmission, which are used to process big data of smart city IoV. Wan et al.[14] proposed an edge computing framework for IoV computing offloading under the 5G network architecture. By simplifying the computing resources to the number of virtual machines and the network resources to a fixed transmission rate, the problem is modelled as a multi-objective optimization problem, and the SPEA2 algorithm is used to solve the offloading scheme.

Edge cloud computing resources are usually limited, so the cloud edge collaborative computing model needs to be considered. Aiming to minimize the total duration of tasks and the cost of server computing resources, Zhao et al.[4] proposed a collaborative method based on MEC and cloud computing to convert the mixed integer programming problem into a computational offloading decision-making problem and a computing resource allocation problem.

Vehicle formation has advantages such as reducing fuel consumption and road congestion and improving road safety, so it is currently the easiest business model for autonomous driving. Cui et al.[15] considered that the members of the vehicle platooning would generate an indivisible computing task in each time slot. They used the Lyapunov optimization algorithm to solve the calculation task offloading strategy. Ma et al.[16] proposed a platoon-assisted vehicle edge computing (PVEC) system, which considers the multi-task offloading system for vehicles was leaving the formation.

Unlike the research mentioned above literature on computing task offloading, this article considers the idle computing resources of other members of the vehicle platooning and the limited computing resources allocated by the MEC server to the vehicle platooning. Our proposed schemes jointly optimize the computing task offloading scheme and the computing resource allocation scheme and aim to minimize the platoon's total energy consumption while considering the urgency of different computing tasks.

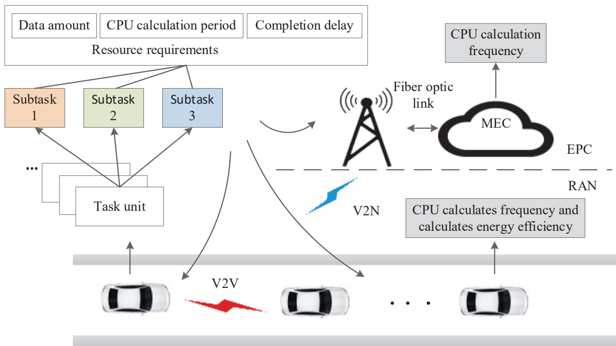


Fig. 1. The scenario of computation offloading in Cooperative Vehicle Infrastructure System.

III. SYSTEM MODEL

Instead of considering the computing resources of RSU and remote cloud, we consider the limited computing resources of the MEC server and other members of the platooning and the limited network resources of the heterogeneous IoV. This paper uses the cooperative vehicle infrastructure scenario of the integration of MEC and IoV to make offloading decisions on the calculation tasks of member vehicles (offloading to MEC server, local calculation, and offloading to another member of the platooning). In addition, this paper allocates the CPU computing resources to optimize the total energy consumption of the vehicle platooning under the premise of satisfying the time delay constraint of the computing task.

In the coverage area of a base station Radio Access Network (RAN) side, there is a vehicle platooning M={m|m=1,2...,M} with M member vehicles travelling on the road. On the Evolved Packet Core (EPC) side, a MEC server (m=0) is deployed near the base station. The member vehicles communicate through Vehicle to Vehicle(V2V), and the vehicles and the base station communicate through Vehicle to Network(V2N). The MEC server communicates with the base station through a high-speed wired optical fibre link. The computing task offloading scenario under a cooperative vehicle infrastructure system is shown in Fig. 1. In this scenario, we set that a MEC server and OBU of M member vehicles can be called a computing unit.

According to the literature[4, 12, 14] and the characteristics of vehicle-infrastructure collaborative computing tasks, this paper proposes a computing task model for vehicle-mounted applications. Each member in the vehicle platooning has a calculation task to be calculated, and we assume that each vehicle requests only one type of calculation task at a time. We define the task unit (the smallest unit of the calculation task that can be divided) of the calculation task of the member vehicle m as a tuple $W_m=\{b_m,\gamma_m,t_m\}$and b_m represents the size of the input data volume of the task unit. We assume that the processing density of the calculation task is uniform. γ_m indicates the processing density of the input data of the task unit, or it can be considered the CPU calculation period required to process unit data. Its value depends on the specific type of vehicle-mounted application to which the computing task tuple belongs. t_m represents the delay constraint of the vehicle-mounted application to which the computing task belongs. This paper considers the urgency of different in-vehicle applications and takes the delay as a constraint to ensure the real-time and reliability of the IoV. In addition, the calculation energy efficiency of each member vehicle is ζ_m, that is, the energy consumed by each CPU calculation cycle of the vehicle to perform the calculation task. Its value depends on the CPU calculation frequency.

Assume that each member vehicle m currently has a batch of tasks to be calculated, where the amount of tasks is α_m. Each vehicle is equipped with two communication antennas, one is used for V2N communication with the base station, and the other is used for V2V communication with other member vehicles. Considering the real-time nature of the vehicle-mounted application of the IoV, the calculation task of the vehicle-mounted application can be divided into three parts for distributed calculation: the number of tasks offloaded from the vehicle m to the MEC server is α_{m0}, the number of local calculation task of vehicle m OBU is α_{m1} and the number of tasks to offload to another vehicle in the platooning is α_{m2}, where α_{mk} is three consecutive integer decision variables and satisfies the following formula:

$$\alpha_m=\sum_{k=0}^{2}\alpha_{mk}\ ,\forall m\in[1,M] \tag{1}$$

A 0-1 integer decision variable of which member vehicle of the platooning is unloaded is c_{ij}, which satisfies the following calculation formula:

$$\begin{cases}\sum_{j=1}^{M} c_{ij}=1,\forall i\in[1,M]\\ c_{ij}=0,\forall i\in[1,M],j=i\end{cases} \tag{2}$$

This article assumes that the bandwidth allocated by the base station to each vehicle fleet is w_0. Because the amount of time of input data transmission is relatively short, during this period, the vehicle's position relative to the base station and the distance between the member vehicles remains approximately unchanged. Therefore, the input data transfer rate is approximately a constant value. The calculation formula for the channel gain of the upload link from the member vehicle m to the base station is[15]:

$$g_{m0}=128.1+37.5\lg(d_{m0}) \tag{3}$$

d_{m0} is the approximate horizontal distance from member vehicle m to the base station, and the calculation formula is:

$$d_{m0}=\sqrt{(x_m-x_0)^2+(y_m-y_0)^2)} \tag{4}$$

where (x_m,y_m) is the abscissa and ordinate information when the member vehicle requests to calculate the task offloading, obtained in real-time by the vehicle-mounted global positioning system (GPS). (x_0,y_0) is the inherent abscissa and ordinate information of the base station. The calculation formula of Signal Noise Ratio (SNR) of the channel is:

$$SNR_{m0}=\frac{p_{m0}g_{m0}}{N_0w_0} \tag{5}$$

where p_{m0} is the transmission power allocated by the base station to the member vehicles, N_0 is the power spectral density of Gaussian white noise, and $N_0+10lgw_0$ is the background noise power. The calculation formula for the data transmission rate of the member vehicle m transmitting the calculation task to the MEC server is:

$$r_{m0}=w_0\log_2(1+SNR_{m0})=w_0\log_2\left(1+\frac{p_{m0}g_{m0}}{N_0w_0}\right) \tag{6}$$

The transmission time and transmission energy consumption for α_{m0} tasks to be offloaded to the MEC server are:

$$\begin{cases} t_{m0}^t=\frac{\alpha_{m0}b_m}{r_{m0}} \\ e_{m0}^t=t_{m0}^t p_{m0} \end{cases} \tag{7}$$

The calculation time spent by the MEC server to calculate α_{m0} tasks from the vehicle m is:

$$t_{m0}^c=\frac{\alpha_{m0}b_m\gamma_m}{f_{m0}} \tag{8}$$

where f_{m0} is the CPU calculation frequency assigned to the vehicle by the MEC server. The total time and total energy consumption for the vehicle m to complete α_{m0} tasks are:

$$\begin{cases} t_{m0}=t_{m0}^t+t_{m0}^c \\ e_{m0}=e_{m0}^t \end{cases} \tag{9}$$

The total time and total energy consumption spent in local calculations are:

$$\begin{cases} t_{m1}=t_{mm}=\frac{\alpha_{m1}b_m\gamma_m}{f_{mm}} \\ e_{m1}=e_{mm}=\alpha_{m1}b_m\gamma_m\zeta_m \end{cases} \tag{10}$$

where f_{mm} is the CPU calculation frequency allocated by vehicle m to α_{m1} tasks of local calculations.

In this article, the bandwidth of the V2V communication mode is w_2. The approximate channel power gain of V2V communication between vehicle m and j is [15]:

$$g_{mj}=63.3+17.7\lg(d_{mj}) \tag{11}$$

The distance d_{mj} between vehicle m and j is:

$$d_{mj}=|m-j|\times d_0 \tag{12}$$

where d_0 is the distance between two adjacent vehicles. The calculation formula for the data transmission rate of the task from vehicle m to j is:

$$r_{mj}=w_2\log_2(1+SNR_{mj})=w_2\log_2\left(1+\frac{p_{m2}g_{mj}}{N_2w_2}\right) \tag{13}$$

where SNR_{mj} is the SNR of the network communication channel between the vehicle m and j, p_{m2} is the transmission power of the vehicle in V2V communication, and N_2 is the power spectral density of Gaussian white noise. The transmission time and transmission energy consumption for α_{m2} computing tasks from vehicle m to j are:

$$\begin{cases} t_{mj}^t=\frac{\alpha_{m2}b_mc_{mj}}{r_{mj}} \\ e_{mj}^t=t_{mj}^t p_{m2} \end{cases} \tag{14}$$

The calculation time and energy consumption of vehicle j to calculate these tasks are:

$$\begin{cases} t_{mj}^c=\frac{\alpha_{m2}b_mc_{mj}\gamma_m}{f_{mj}} \\ e_{mj}^c=\alpha_{m2}b_mc_{mj}\gamma_m\zeta_j \end{cases} \tag{15}$$

where f_{mj} is the CPU calculation frequency allocated by vehicle j for these tasks. The total time and total energy consumption to complete these tasks are:

$$\begin{cases} t_{mj}=t_{mj}^t+t_{mj}^c \quad t_{m2}=\sum_{j=1,j\neq i}^{M}t_{mj} \\ e_{mj}=e_{mj}^t+e_{mj}^c \quad e_{m2}=\sum_{j=1,j\neq i}^{M}e_{mj} \end{cases} \tag{16}$$

Based on ensuring the service quality of the vehicle platooning, this article aims to minimize the total battery energy consumption. The energy consumption of each member vehicle includes the CPU calculation energy consumption for local calculation, transmission energy consumption of V2N communication mode for offloading to the MEC server, transmission energy consumption of V2V communication mode for offloading to other member vehicles, and the corresponding CPU calculation energy consumption. The calculation formula is as follows:

$$\min f_0(c_{ij},\alpha_{ik},f_{ij})=\sum_{i=1}^{M}\sum_{k=0}^{2}e_{ik} \tag{17}$$

In this paper, vehicle-infrastructure collaborative computing task offloading is modelled as an integer nonlinear programming model. The specific calculation formula is as follows:

$$\begin{cases} \text{obj.}\ \min f_0(c_{ij},\alpha_{ik},f_{ij}) \\ \text{s.t.}C1:\sum_{i=1}^{M}f_{ij}-f\leq 0_j \\ C2:\max(t_{ik})\leq t_i \\ C3:\alpha_{ik}\geq 0,\sum_{k=0}^{2}\alpha_{ik}=\alpha_i,\alpha_{ik}\in Z \\ C4:\sum_{j=1}^{M}c_{ij}=1,c_{ii}=c_{i0}=0,c_{ij}=\{0,1\} \\ C5:\begin{cases} i=\{1,m,\ldots,M\} \\ j=\{0,m,\ldots,M\} \\ k=\{0,1,2\} \end{cases} \end{cases} \tag{18}$$

where formula C1 means that the sum of the CPU calculation frequency assigned to the calculation tasks offloaded to the same calculation unit cannot exceed the calculation capacity of the calculation unit, formula C2 indicates that tasks that are offloaded to different computing units need to be calculated within their specified delay constraints, formula C3 indicates

TABLE I. THE PSEUDO-CODE OF THE EALNS ALGORITHM

```
Extended Adaptive Large Neighbourhood Search
Input: An initial solution x_0 of computation
    offloading problem.
Output: A better solution x* of computation
    offloading problem.
Initialization: x_glo=x_0; x=x_0; ω- = [1,...,1];
    ω+ = [1,...,1]
Repeat
select destroy and repair methods d_u'(·)∈Ω- and
    r_v'(·)∈Ω+ using ω- and ω+;
x_cur=r_v'(d_u'(x));
If accept(x_cur,x) Then
    x=x_cur;
End If
If f(x_cur)<f(x_glo) Then
    x_glo=x_cur;
End If
update ω- and ω+;
until stopping criterion is met
```

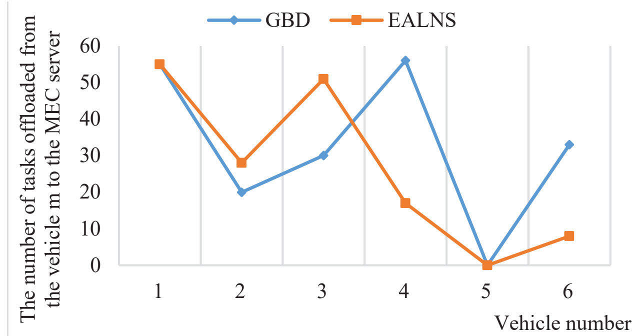
that the amount of tasks offloaded to different computing units is a non-negative integer, and the sum is equal to the total amount of tasks, formula C4 indicates that for each calculation task, a member vehicle other than itself must be selected.

The pseudo-code of the extended adaptive large neighbourhood search (EALNS) algorithm is shown in Table I, where x_{glo} means the current optimal solution obtained during the iteration, x means the current solution, x_{cur} represents the temporary solution obtained by the current solution x after a removal heuristic and a repair heuristic. According to the acceptance criterion accept(x_{cur},x), it is judged whether to use the temporary solution x_{cur} as the current solution x in the next iteration. By comparing $f(x_{cur})$ with $f(x_{glo})$, judge whether to use x_{cur} as the current optimal solution x_{glo} in the next iteration. $d_{u'}(\cdot)$ and $r_{v'}(\cdot)$ respectively represent the removal heuristic and repair heuristic selected in an iteration process. Ω^- and Ω^+ are the set of removal heuristics and repair heuristics proposed in this article. ω^- and ω^+ are the weight sets of each removal heuristic and repair heuristic respectively.

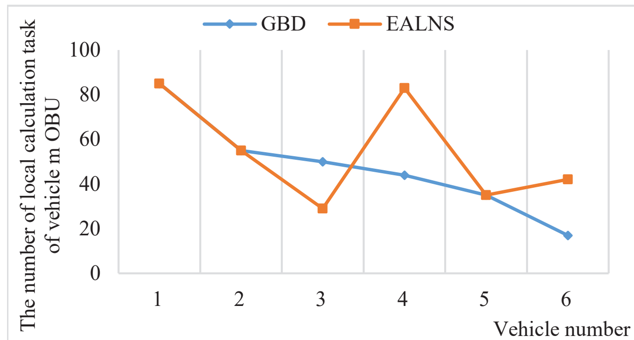
IV. SIMULATION RESULTS

In this section, we present simulation results to evaluate the performance of the proposed algorithms. To facilitate the evaluation, we compare the algorithm proposed in this article with the generalized Benders decomposition algorithm (GBD), an accurate algorithm for solving mixed-integer nonlinear programming (MINLP).

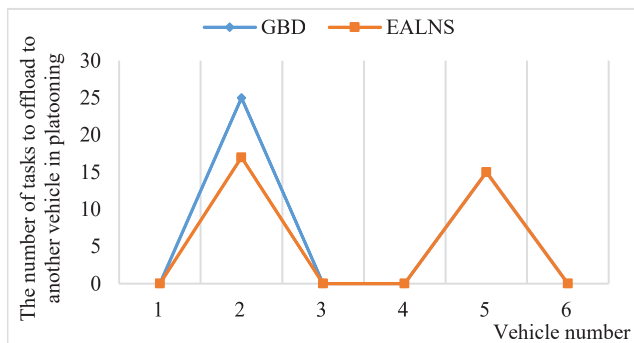
Simulation environment configuration is as follows: Hardware is Intel Core i5-9300HF CPU. The operating system is 64-bit Window 10 Professional Edition. The development tools are Java SE Development Kit 8u281, eclipse-committers-2021-03-R-win32-x86_64, and EdgeCloudSim.



(a) α_{m0}



(b) α_{m1}



(c) α_{m2}

Fig. 2. The computing task offloading schemes obtained by GBD algorithm and EALNS algorithm.

TABLE II. THE CHARACTERISTICS OF COMPUTING TASKS ON VEHICLES

	1	2	3	4	5	6
t_m (ms)	350	250	500	300	400	450
α_m	140	100	80	100	50	50
b_m (kb)	16	20	25	12	32	24
γ_m (10^2 cycles/bit)	2.5	2.25	3.75	3	3.5	2.25

When the number of team members is 6, each member's vehicle calculation task characteristics are shown in Table II. The computing task offloading schemes obtained by GBD algorithm and EALNS algorithm are shown in Fig. 2.

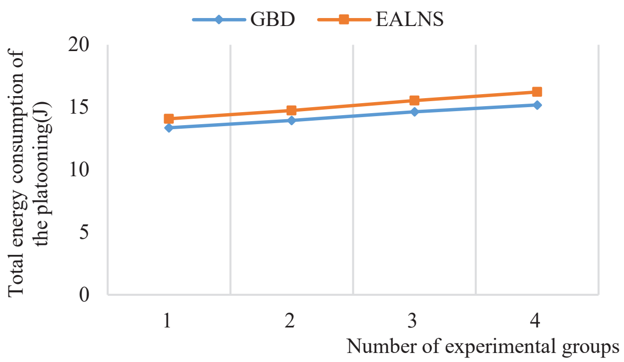


Fig. 3. The comparison of energy consumption values obtained by the two algorithms.

Then, when the number of team members is 6, we set up four sets of experiments in which the task volume and complexity of the calculation task increase in sequence. The comparison of the total energy consumption of the platooning of the two algorithms is shown in Fig. 3.

It can be observed from Fig. 3 that as the amount and complexity of the calculation tasks of the member vehicles increase, the unfeasible single-vehicle calculation task offloading schemes generated by the EALNS algorithm in the solution process increase. However, compared with the optimal total energy consumption of the platooning obtained by the GBD algorithm, this meta-heuristic algorithm is still effective. It increases the difficulty of getting closer to the optimal solution and slows down the search process.

V. Conclusion

In this paper, we investigate the task offloading mechanisms in vehicular edge computing networks to minimize the total energy consumption of the platooning while the task completed maximum latency and other constraints are satisfied. We analyze the model of the edge computing task offloading scheme in the vehicle platooning scenario. When many tasks arrive, the vehicle can send part of them to the MEC servers and another member vehicle. Moreover, the EALNS algorithm is proposed. Extensive simulations are proposed to demonstrate the effectiveness of the EALNS algorithm.

References

[1] K. Sasaki, N. Suzuki, S. Makido, A. Nakao, and Ieee, Vehicle Control System Coordinated Between Cloud and Mobile Edge Computing (2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan). 2016, pp. 1122-1127.

[2] W. Lu, Y. Gong, X. Liu, J. Wu, and H. Peng, "Collaborative Energy and Information Transfer in Green Wireless Sensor Networks for Smart Cities," Ieee Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1585-1593, Apr 2018.

[3] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar, "Dependable Demand Response Management in the Smart Grid: A Stackelberg Game Approach," Ieee Transactions on Smart Grid, vol. 4, no. 1, pp. 120-132, Mar 2013.

[4] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 8, pp. 7944-7956, 2019.

[5] A. A. Alahmadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Distributed Processing in Vehicular Cloud Networks," in Proceedings of the 2017 8th International Conference on the Network of the Future, T. Mahmoodi, S. Secci, A. Cianfrani, and F. Idzikowski, Eds. (International Conference on the Network of the Future, 2017, pp. 22-26.

[6] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward Cloud-Based Vehicular Networks with Efficient Resource Management," Ieee Network, vol. 27, no. 5, pp. 48-55, Sep-Oct 2013.

[7] M. Nabi, R. Benkoczi, S. Abdelhamid, H. S. Hassanein, and Ieee, "Resource Assignment in Vehicular Clouds," in 2017 Ieee International Conference on Communications(IEEE International Conference on Communications, 2017.

[8] Y. Sun et al., "Adaptive Learning-Based Task Offloading for Vehicular Edge Computing Systems," Ieee Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3061-3074, Apr 2019.

[9] X. Xu et al., "An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles," Future Generation Computer Systems-the International Journal of Escience, vol. 96, pp. 89-100, Jul 2019.

[10] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks," IEEE Access, vol. 7, pp. 26652-26664, 2019.

[11] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, Delay Constrained Offloading for Mobile Edge Computing in Cloud-enabled Vehicular Networks (Proceedings of 2016 8th International Workshop on Resilient Networks Design and Modeling). 2016, pp. 288-294.

[12] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4377-4387, 2019.

[13] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management," IEEE Communications Magazine, vol. 55, no. 12, pp. 60-67, 2017.

[14] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, "Efficient computation offloading for Internet of Vehicles in edge computing-assisted 5G networks," Journal of Supercomputing, vol. 76, no. 4, pp. 2518-2547, Apr 2020.

[15] T. Cui, Y. Hu, B. Shen, and Q. Chen, "Task Offloading Based on Lyapunov Optimization for MEC-Assisted Vehicular Platooning Networks," Sensors, vol. 19, no. 22, Nov 2019, Art. no. 4974.

[16] M. Xiaoting, Z. Junhui, L. Qiuping, and G. Yi, Reinforcement learning based task offloading and take-back in vehicle platoon networks (2019 IEEE International Conference on Communications Workshops). 2019, pp. 6 pp.-6 pp.