Noname manuscript No.
(will be inserted by the editor)

Efficient Computation Offloading for Internet of Vehicles in Edge Computing-Assisted 5G Networks

Shaohua Wan · Xiang Li · Yuan Xue · Wenmin Lin · Xiaolong Xu*

Received: date / Accepted: date

Abstract The Internet of Vehicles (IoV) is employed to gather real-time traffic information for drivers, and base stations in 5G systems are used to assist in traffic data transmission. For rapid implementation, the applications in vehicles are available to be offloaded to edge nodes (ENs) which are enhanced from micro base stations. Despite the benefits of IoV and ENs, the explosive growth of offloaded vehicle applications exceeds the capacity of ENs, caus-

Shaohua Wan
School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, Hubei, China.
E-mail: shaohua.wan@ieee.org

Xiang Li
School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China.
E-mail: nuistlixiang@gmail.com

Yuan Xue
School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China.
E-mail: xueyuannuist@gmail.com

Wenmin Lin
School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China.
Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou 310018, China.
E-mail: linwenmin@hdu.edu.cn

Xiaolong Xu*
School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China.
E-mail: njuxlxu@gmail.com
*Xiaolong Xu is the corresponding author.

ing the overload of fractional ENs. Therefore, it is necessary to offload the computing applications in overloaded ENs to other idle ENs while it is a challenge to select appropriate offloading destination ENs. In this paper, we first consider edge computing framework for computation offloading in IoV under the architecture of 5G networks. We then formulate a multi-objective optimization problem to select suitable destination ENs, which aims to minimize the vehicle application offloading delay and offloading cost as well as realizing the load balance of ENs. Moreover, a computation offloading method for IoV, named COV, is designed to solve the multi-objective optimization problem. Finally, various simulation analyses demonstrate the effectiveness and efficiency of COV.

Keywords IoV · 5G networks · edge computing · computation offloading · delay · offloading cost · load balance

1 Introduction

1.1 Background

In recent years, a growing number of vehicles occur on the road [1]. Considering that there are numerous vehicles on the road, the probability of traffic accident rises sharply, endangering people's personal safety. Additionally, provided that people encounter traffic jams on the road, plentiful time is wasted on useless waiting, giving drivers and passengers who are queueing a poor experience[2]. Therefore, it is an urgent demand to manage traffic conditions to reduce accidents and congestions as well as saving valuable time for people. Internet of Vehicles (IoV) has emerged as an appropriate paradigm, which aims to make drivers obtain real-time traffic information [3]. Generally, the real-time traffic information includes the overall traffic management and the intricate movement of vehicles since the IoV environment combines road conditions and wireless networks [3][4][5]. In the IoV environment, vehicles are equipped with intelligent devices such as sensors and actuators that gather information from surrounding conditions. Those intelligent devices are connected with roadside units (RSUs) which is in charge of data receiving and sending [6]. Consequently, once the drivers need non-local traffic information, real-time road condition data are transmitted from RSUs to the destination vehicle fleetly, achieving the information interaction amongst vehicles, humans and RSUs [3][7].

In addition, by means of vehicle-to-infrastructure (V2I) communication, autonomous vehicles acquire traffic data from RSUs to realize adaptive cruise control and route guidance systems, functioning as fully automated driving [8]. However, traffic data are too massive to be processed in time and the data transmission may be discontinuous. As autonomous vehicles are intolerant of service interruptions, if there is an interruption in traffic data transmission, autonomous vehicles miss real-time road conditions so that incomplete traffic data analysis is conducted, resulting in unpredicted accidents. For the sake of driving safety, the fifth-generation (5G) networks are potential to be a solution. In 5G networks, millimeter wave (mmWave), whose bands are above 10 GHz, is adopted for data transmission [9][10][11]. Taking advantage of vast unused spectrum, mmWave offers a high data transmission rate, which is orders of magnitude higher than the data transmission rate in 4G systems [12][13]. Besides, base stations exist in 5G systems, which are in reality radio transceiver stations that communicate with various terminal devices. By means of mmWave, the base stations are promising to connect everything with a rapid data transmission rate [14]. In view of the efficient data process in 5G networks, it is suitable to combine 5G networks and IoV, especially in autonomous driving.

According to geographic distributions of base stations, each base station has its own coverage and deals with computing applications offloaded from the vehicles in the coverage [15]. Nevertheless, the processing power of base stations is insufficient to meet the state of the art requirements of complex computing applications, making a drop in the quality of experience (QoE) for drivers and passengers. Thus, the applications are offloaded to the centralized cloud data centers for processing while the long distance between vehicles and cloud platform contributes to low transmission efficiency and unstable connections [16]. Edge computing, which pushes the computing services to the edge of the radio network, emerges to be a complement to 5G networks [17][18]. As the base stations in 5G systems are divided into macro base stations (MABSs) and micro base stations (MIBSs), MABSs have a wide coverge area while MIBSs have a relatively narrow coverage. There are many infrastructures that can be enhanced as edge computing devices, including RSUs, MABSs and MIBSs [19]. Considering that the mobile applications are usually under the coverage of MIBSs, edge devices are co-located with MIBSs and MIBSs are enhanced to be edge nodes (ENs) to assist MABSs in data processing as the result. The computing applications of vehicles are offloaded to ENs that are in close range to vehicles. As there are physical servers deployed in each EN, via offloading computing applications from vehicles to ENs, the application offloading time is reduced and the QoE for drivers and passengers in executing computing applications is greatly improved. Whereas in reality, when the data processing capability of ENs cannot satisfy drivers' requirements, it is wise to offload the computing applications in ENs to corresponding MABS for better implementation effect. Moreover, specific computing applications, which need designated data from cloud data centers, are offloaded to the cloud platform from MABS, represented by video stream acquisition in vehicles [20].

Nevertheless, in actual operation, ENs may be overload during peak hours. On account of the massive offloaded applications, the computing resources and storage in the EN are all occupied. Thereby, subsequent offloading requests from vehicles are rejected and queued in the EN until previous applications are completed and the required resources and storage are available again. Thereby, the application execution efficiency is terrible, decreasing the QoE for drivers and passengers. It is of necessity to offload the queueing applications to another EN which has rich idle computing resources [21][22]. Nonetheless, deciding which EN to be the offloading destination must be considered comprehensively. During the process of offloading applications from one EN to another, a mass of extra data transmission delay is produced [23]. Obviously, the data transmission delay depends on the location of origin EN and the selection of destination EN. Furthermore, as virtualized technique is used for data processing, the load balance of VMs in ENs is taken into account to strengthen the data processing capabilities of ENs. From the perspective of the service providers, the offloading cost is an emphasis, which is influenced by the amount of offloaded applications from vehicles. In this situation, the offloading strategies are formulated due to the comprehensive consideration of offloading delay, offloading cost and load balance of ENs.

1.2 Related Work and Motivation

With the advances in Wireless Sensor Networks (WSN) and Internet technologies, IoV emerges as a key technical field to provide real-time traffic information for drivers [24][25][26]. In IoV environment, there is abundant traffic information to gather and process, decreasing the efficiency of application execution in vehicles sharply [27][28][29]. In terms of using computing devices which are in close proximity to vehicles to assist in data processing, edge computing emerges as a suitable solution [30][31]. In the field of edge computing, Zhao et al. focused on the reduction of energy consumption and designed an algorithm to optimize the transmission power of application offloading between vehicles and RSUs [32]. In [33], Baktir et al. proposed a model which integrate software-defined networking (SDN) into edge computing, and deduced a group of "Benefit Areas" based on the functions of SDN. In smart grids, Kumar et al. adopted edge computing for data dissemination to smart devices and comprehensively analyzed response time, data transmission delay, and throughput to the end terminals when using vehicles as mobile nodes [34]. In [35], Zhu et al. considered using edge computing to optimize the web. After the proof of the experiment, the approach was proved effective.

Although many studies have derived schemes for application offloading in edge computing, in toward 5G networks, the traditional edge computing framework is of low efficiency to process massive applications. That is to say, with the explosive growth of computing applications and the demanding requirements from drivers, it is inappropriate to adopt the traditional edge computing architecture in 5G networks. In [36], Zhang et al. studied the computation offloading mechanisms for edge computing in the aspect of energy consumption. Based on the multi-access characteristics in 5G heterogeneous network, they proposed an offloading method to optimize radio resource allocation to minimize the energy consumption under the latency constraints. From the perspective of Device-to-Device (D2D) communication, Chen et al. focused on the energy-efficient D2D Crowd system in which numerous devices are interconnected at the edge of 5G networks for data communication and cooperation. With the help of 5G network-assisted D2D collaboration, they proposed an application allocation method based on graph matching technology to achieve the goal of saving energy. Besides, they extended the D2D Crowd framework to adapt various applications in reality [37]. On the other hand, Guo et al. formulated the distributed computation offloading problem as a potential game, which is proved to satisfy the Nash equilibrium. According to the potential game, they designed an offloading algorithm to jointly optimize the latency and energy consumption of smart devices in IoV [38]. Taking the unique property of 5G networks into consideration, Yang et al. designed an artificial fish swarm algorithm to minimize the energy consumption of all entities [39]. Futhermore, Bastug et al. verified that increasing the storage capability of edge devices has the ability to effectively ease the network congestion [40]. Considering that data transmission is carried over fronthaul and backhaul links, the goals of the algorithm include service delay requirement and computation capabilities. In this paper, several physical servers are placed around each MIBS, which is known as EN, enhancing the performance of implementing the computing applications in vehicles. Nevertheless, the computation offloading process is restricted considering that the ENs may be overloaded. Under this circumstance, the new requests from vehicles are rejected and queued in the EN until the required resources are available. Therefore, deciding which EN the waiting computing applications are offloaded to depends on the comprehensive consideration of application offloading delay, application offloading cost and the load balance of ENs.

1.3 Paper Contributions

We summarize the main contributions of this paper as follows:

- We analyze the process of computation offloading across ENs and formulate the computation offloading problem as a multi-objective optimization problem to minimize the application offloading delay and offloading cost while realizing the load balance of ENs.
- We employ strength pareto evolutionary algorithm 2 (SPEA2) to draw up several executable computation offloading schemes.
- We adopt the technique for order preference by Similarity to an Ideal Solution (TOPSIS) and multi-criteria decision-making (MCDM) to select appropriate computation offloading schemes.
- We formally demonstrate the effectiveness and efficiency of our proposed algorithm by simulation experiments.

The remainder of this paper is organized as follows. In Section 2, we introduce the IoV computation offloading in edge computing paradigm under the architecture of 5G networks. In Section 3, we model the complete computation offloading problem in terms of minimizing offloading delay and offloading cost as well as achieving the load balance of ENs. In Section 4, we present a computation offloading method for IoV in edge computing. In Section 5, we provide simulation experiments to evaluate the performance of the proposed method. In Section 6, we outline the conclusions and future work.

2 IoV Offloading in 5G Networks with Edge Computing

The major contribution of the forthcoming 5G network is to realize the vision of connecting everything by the base station by taking advantage of the mmWave. In the 5G wireless communication, there are multiple MABSs deployed for providing services for the IoT applications, e.g., D2D communications and vehicle-to-vehicle (V2V) applications [41]. The MABS covers multiple MIBSs to enhance the service quality and they can receive various service requests from the end equipment with wireless signals, such as the wearable devices, the vehicles, the mobile phones, etc. Therefore, the IoV applications could be offloaded through the 5G networks.

In the traditional wireless communication, the IoV applications are offloaded for processing through RSUs to the base station. If the computing resources in the base station are scarce for application execution, these applications are transferred from the base station to the remote cloud platform for processing. However, in the 5G communication, the vehicle applications are transferred to the MIBSs first from the RSUs, and then they are transferred to the MABSs. To enhance the service performance for implementing the vehicle applications, we employ ENs to assist in processing applications under the unique architecture of 5G networks.

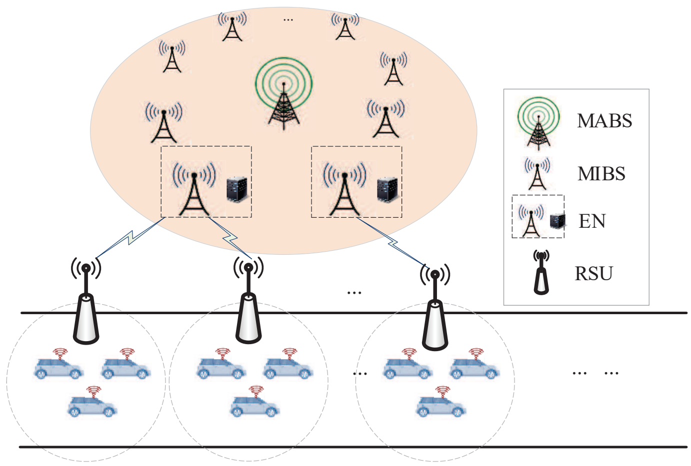


Fig. 1: Edge-enabled communication framework for IoV in 5G networks.

In our consideration, there are three infrastructures that can serve as ENs, which are RSUs, MIBSs, and MABSs. Nevertheless, on one hand, RSUs merely receive and process the applications offloaded from vehicles, which are too unitary to work in the whole 5G networks. On the other hand, the coverage of the MABS is extremely wide, contributing to the low data transmission speed and unstable data transmission connection. Therefore, we configure MIBSs as

ENs by placing the physical servers around each MIBS, as shown in Fig. 1. These ENs are distributed densely in the urban cities around the road, thus the ENs could be leveraged efficiently for accommodating the vehicle applications. ENs provide real-time edge services to the vehicle applications rather than engaging the remote cloud platforms for application implementation with high transmission delay.

The computing capacity of the EN is often limited by its physical size and deployment context of the MIBS, thus the EN may receive many application offloading requests and become overload during peak hours in IoV. In view of the load condition of all ENs, it is essential to conduct efficient resource provisioning for the ENs while responding to the offloading requests from the vehicle application. For resource provisioning from the ENs, the virtualized techniques, like Xen, VMWare, Hyper-V etc., can be employed, since they are proved to be efficient for resource management for the cloud infrastructure. By leveraging the virtualized technique, we can simplify the resource allocation problem for offloading processes as the VM placement and VM scheduling.

In the process of application offloading, the vehicle first sends the offloading request to the proximate RSU and transmits the application to the destination RSU once receiving offloading permission. As the resources in the RSU are limited, for better application processing results, the application in the RSU are transmitted to the EN from the RSU. Provided that the processing demanded by the application is too complicated, the application is transmitted to MABS which has massive computing resources. For those applications whose processing need specific data in data centers, the applications are transmitted to the cloud platform for execution. Besides, terribly in the process of application offloading, the EN may be overload and the application is queued in the EN until the required resources are available. To achieve efficient offloading, the application is offloaded to another idle EN for processing.

3 System Model

In this section, the system model of computation offloading for IoV in Edge Computing-Assisted 5G networks is proposed and the computation offloading problem is formulated as a multi-objective optimization problem. Key notations are demonstrated in Table 1.

3.1 Resource Model

In the 5G network, note $S = \{s_1, s_2, \ldots, s_N\}$ as the MIBS set, where N is the amount of the MIBSs deployed around a MABS. The MIBSs are enhanced as the edge nodes (ENs) for responding the resources requests of the computing applications. The extended EN set is denoted as $E = \{e_1, e_2, \ldots, e_N\}$. As the virtualized techniques are introduced for the management of the ENs, virtual machines (VMs) are served as the computing units. Thus, the maximum

Table 1: Notation

Notation	Definition
N	The amount of the MIBSs deployed around a MABS
S	The MIBS set, $S = \{s_1, s_2, \ldots, s_N\}$
E	The extended EN set, $E = \{e_1, e_2, \ldots, e_N\}$
e_n	The n-th EN in E
α_n	The capacity of e_n
M	The number of the vehicles along the road
V	The vehicle set, $V = \{v_1, v_2, \ldots, v_M\}$
VA	The vehicle application set, $VA = \{va_1, va_2, \ldots, va_M\}$
va_m	The n-th vehicle application in VA
θ_m	The amount of VMs for implementing va_m
D	The offloading delay for all applications along the road
C	The total transmission cost for all ENs
L	The load balance degree for all ENs

capacity of ENs is determined by the quantity of the containable VMs, and the capacity of $e_n(n = \{1, 2, ..., N\})$ is record as α_n.

The vehicles running along the road surrounded by massive RSUs and dense MIBSs need the feedback for the implementation of the IoV applications. To win the instant gratification on QoE of the vehicle users, the IoV applications are committed to being deployed with one or more VMs. Note the vehicle set as $V = \{v_1, v_2, \ldots, v_M\}$, where M is the number of the vehicles along the road. Without loss of generality, suppose that each vehicle submit one IoV application to be offloaded. Correspondingly, the vehicle application set is denoted as $VA = \{va_1, va_2, \ldots, va_M\}$. The amount of quantificational VMs for implementing the application va_m $(m = \{1, 2, \ldots, M\})$ is denoted as θ_m.

3.2 Offloading Delay Model

For the application va_m, if it is offloaded to an EN for implementation, the transmission delay is consisted of the transmission time between the vehicle and the nearby RSU, the transmission delay between RSU and the accessible EN, as well as the transmission delay between the accessible EN and the goal EN, as shown in Fig. 2. Otherwise, when the computing application is offloading to the cloud platform for execution, the communication delay should take the transmission time between the EN to the cloud platform into account. The implementation results should return back from the implantation node, i.e., EN or the cloud platform, to the moving vehicle that publishes the application.

The offloading time between v_m and the nearby RSU, denoted as VR_m, is determined by

$$VR_m = \frac{s_m}{tr_{VR}}, \tag{1}$$

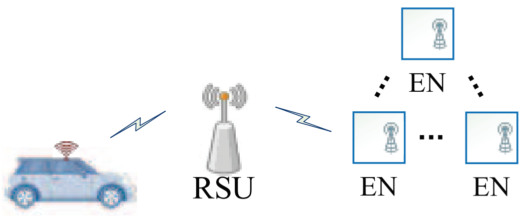


Fig. 2: An example of computation offloading in edge computing.

where s_m is the occupied memory size of va_m and tr_{VR} is the transmission rate between the vehicle and the RSU.

Note the transmission time for transferring va_m from the RSU to the accessible EN with a MIBS as RE_m, which is calculated by

$$RE_m = \frac{s_m}{tr_{RE}}, \tag{2}$$

where tr_{RE} is the transmission rate between the RSU and the accessible EN.

When offloading va_m from the accessible edge node to the goal edge node, va_m may need to transfer through one or more ENs. Note the amount of the transferred ENs as γ_m. And the offloading time is determined by the amount of the ENs that va_m transfers through. Thus the transfer time denoted as GE_m is calculated by

$$GE_m = \begin{cases} 0, & \gamma_m = \{0, 1\}, \\ \frac{s_m}{tr_{EN}}.(\gamma_m - 1), & \gamma_m > 1, \end{cases} \tag{3}$$

where tr_{EN} is the transmission rate between two ENs.

Then the offloading delay of va_m is determined by

$$OD_m = VR_m + RE_m + GE_m. \tag{4}$$

Correspondingly, the offloading delay for all the applications along the road is calculated by

$$D = \sum_{m=1}^{M} OD_m. \tag{5}$$

3.3 Offloading Cost Model

The offloading cost for the service provider of 5G mainly refers to the transmission amount of the vehicle applications for ENs due to receiving and transferring the vehicle applications. The transmission data for computation offloading among ENs and MIBS affect the 5G cost for the service providers.

Fig. 3 shows an example of data transmission with γ_m= 0, γ_m= 1 and γ_m= 2. When γ_m= 0, the application is offloaded to the destination EN directly,

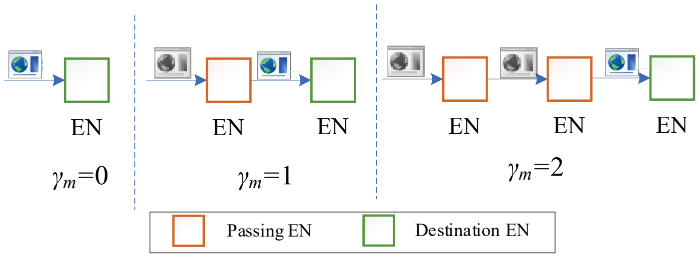


Fig. 3: An example of data transmission with γ_m= 0, γ_m= 1 and γ_m= 2.

thus the total amount of the transferred data is s_m. When γ_m= 1, there is one passing EN that the av_m should be transferred through, and the total amount of the transferred data is by analogy, the total transmission cost for all the ENs, denoted as C, is determined by

$$C = \sum_{m=1}^{M} (s_m + 2\gamma_m \cdot s_m) = \sum_{m=1}^{M} (2\gamma_m + 1) \cdot s_m. \tag{6}$$

3.4 Load Balance Model

The load balance of ENs refers to the fair degree for the load distribution of the offloaded vehicle applications. As described in 3.1, the vehicle application acquires θ_m VM instances in ENs for execution, and to reduce the data interactions among the ENs for implementing va_m, these VM instances should be provisioned from the same EN [42].

Note P_m^n as a flag to record the distribution location of va_m determined by

$$P_m^n = \begin{cases} 0, & va_m \text{ is offloaded to } e_n, \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

The balanced degree of ENs relies on the load distribution for the vehicle applications among the ENs. The utilized VMs for implementation on e_n, denoted as z_n, is calculated by

$$z_n = \sum_{m=1}^{M} P_m^n \cdot \theta_m. \tag{8}$$

Then like the analysis of utilization for cloud servers, the utilization for the EN e_n, denoted as u_n, is calculated by

$$u_n = z_n / \alpha_n. \tag{9}$$

Note Q_n as the flag to decice whether e_n is leverage for accommodating the vehicle applications, which is measured by

$$Q_n = \begin{cases} 1, z_n \neq 0, \\ 0, z_n = 0. \end{cases} \tag{10}$$

The amount of utilized ENs is calculated by

$$W = \sum_{n=1}^{N} Q_n. \tag{11}$$

Then the average utilization for the leveraged ENs is determined by

$$U = \frac{1}{W} \sum_{n=1}^{N} u_n. \tag{12}$$

The load deviation for the load distribution of e_n, denoted as ld_n, is calculated by

$$ld_n = |u_n - U| \,. \tag{13}$$

Then the load balance degree for all the ENs is defined by

$$L = \frac{1}{W} \sum_{n=1}^{N} \frac{1}{ld_n}. \tag{14}$$

3.5 Problem Definition

After the model construction of offloading delay, offloading cost and load balance for migrating the vehicle applications to ENs in 5G networks, our problem is defined by

$$\min D, \, \min C, \, \max L. \tag{15}$$

$$s.\ t. \quad \sum_{m=1}^{M} P_m^n \le \alpha_n, \tag{16}$$

$$s.\ t. \quad \sum_{n=1}^{N} P_m^n = 1, \tag{17}$$

$$s.\ t. \quad 0 \le \sum_{n=1}^{N} Q_n \le N. \tag{18}$$

4 An Efficient Computation Offloading method for Internet of Vehicles in Edge Computing-Assisted 5G Networks

The purpose of this paper is to solve a multi-objective optimization problem which aims at reducing the offloading delay, promoting the offloading cost and optimizing the load balance at the same time. Compared with traditional algorithms such as weighted coefficient method and genetic algorithm (GA), SPEA2 is widely used in multi-objective optimization problems because of its good robustness, simple and universal, global optimization and parallel processing mechanism. So SPEA2 is selected to solve our problem.

4.1 Determination of the destination EN

In this paper, the ENs are not empty at the beginning. There are already a quantity of vehicle applications in the ENs, so when the vehicle applications are offloaded to a start EN full of vehicle applications, the vehicle applications will be offloaded to the next EN until the VMs of the next EN can implement the vehicle applications. Fig. 4 shows an example of obtaining the destination EN.

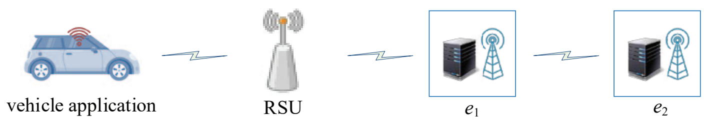


Fig. 4: An example of obtaining the destination EN.

In Fig. 4, the vehicle application is transmitted to the nearby RSU. Then the application is offloaded to e_1 at first, but the vacant VMs of e_1 are less than the VMs which are needed to implement the application. Therefore, the vehicle application is transferred from e_1 to e_2, because e_2 has enough VMs to execute the vehicle application.

In the following Algorithm 1, we elaborate the procedure of looking for the destination EN denoted by e_D, and e_S represents the start EN which is the first EN RSU offloads to.

4.2 Encoding

In the encoding operation, all of the ENs should be encoded at first. In the GA, each EN which is transferred by the vehicle applications at last should be represented as a gene. All the genes make up the chromosome, which represents the efficient computation offloading strategy for the vehicle applications.

Algorithm 1 Obtain e_D

Require: E, va_m
Ensure: e_D

```
obtain e_S
for i = S (S is the subscript of e_S) to N do
    if α_N < θ_m then
        i=i+1
    else
        break
    end if
end for
D = i
return e_D
```

Fig. 5 gives an example of efficient computation offloading strategy, and the chromosome is encoded in integer in this paper.

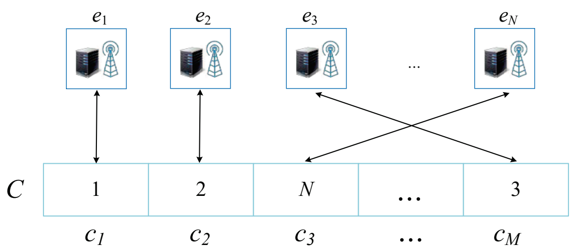


Fig. 5: An example of computation offloading strategy.

4.3 Fitness Functions and Constraints

In GA, the fitness functions are used to make an evaluation of each individual's pros and cons, then the opportunity of inheritance can be determined. In this paper, the fitness functions include three categories: the offloading delay (5), the offloading cost (6) and the load balance degree (14). As is shown in (15), the purpose of this method is to reduce the offloading delay, promote the offloading cost and optimize the load balance at the same time. The constraint is given by (16), representing the need of VMs of the vehicle applications is less than the vacant VMs of EN.

In the Algorithm 2, the offloading delay for the vehicle can be evaluated at last. In this algorithm, we should input M. The offloading time between v_m and the nearby RSU, the transmission time for transferring va_m from the RSU to the accessible EN with a MIBS and the transfer time can be obtained (Lines 2-5). Finally, the offloading delay for all the applications is equal to the summation of VR_m, RE_m, GE_m (Line 6).

Algorithm 2 Offloading delay evaluation for vehicle applications

Require: M
Ensure: D
for $m = 1$ to M **do**
    Calculate VR_m by formula (1)
    Calculate RE_m by formula (2)
    Obtain γ_m by Algorithm 1
    Obtain GE_m by formula (3)
    D+=VR_m+RE_m+GE_m
end for
return D

In Algorithm 3, the transmission cost can be evaluated at last. In this algorithm, we should input M, s_m, γ_m. The total transmission cost for all the ENs can be evaluated (Lines 1-5).

Algorithm 3 Transmission cost evaluation for all the ENs

Require: M, s_m, γ_m
Ensure: C
1: **for** $m = 1$ to M **do**
2:     Calculate C_m by formula (6)
3:     C+=C_m
4: **end for**
5: **return** C

In the Algorithm 4, the load balance degree for all the ENs can be evaluated at last. In this algorithm, we should input M, N, θ_m and α_n. The utilization for the EN e_n should be evaluated at first (Lines 2-5). Then the amount of utilized ENs can be calculated (Lines 7-12) w and u_n are used to calculate the average utilization for the leveraged ENs and the load deviation for the load distribution of e_n (Lines 13-16). At last, the load balance degree for all the ENs is calculated by W, N and ld_n (Line 17).

4.4 Initialization

In the initialization operation, the paraments should be determined at first, including the size of population S, the size of archive A, the number of iterations T, the probability of crossover E and the probability of mutation Y.

Each chromosome represents the M efficient computation offloading strategies of the M computing applications, the strategy can be denoted as $C_\lambda(c_{\lambda,1}, c_{\lambda,2}, c_{\lambda,3}, ..., c_{\lambda,m})$, where C_λ represents the λ-th chromosome and $C_{\lambda,j}$ represents the j-th gene of the λ-th chromosome.

Algorithm 4 Load balance degree evaluation for all the ENs

Require: M, N, θ_m, α_n
Ensure: L

```
for m = 1 to M do
    if P_m^n=1 then
        Calculate Z_n by formula (8)
        Calculate u_n by formula (9)
    end if
end for
for n = 1 to N do
    if z_n ≠ 0 then
        Q_n=1
    else
        Q_n=0
        W_n=Q_n
        W+=W_n
        U_n=u_n/w
        U+=U_n
        Calculate ld_n by formula (13)
        L_n=1/(w*ld_n)
        L+ = L_n
    end if
end for
return L
```

4.5 Selection

In the selection operation, the individuals with high fitness are selected from the current evolutionary group into the mating pool. The crossover operation and the mutation operation can only select individuals from the mating pool to generate a better population.

In the SPEA2 algorithm, the method of calculating individual's fitness $G(k)$ is improved, taking the nondominated individuals and the dominated individuals into account at the same time, as calculated by:

$$V(k) = \sum_{l \in Q_t + P_t, l \succ k,} H(l), \tag{19}$$

$$H(k) = |\{l | l \in Q_t + P_t \wedge k \succ l\}, \tag{20}$$

$$B(k) = \frac{1}{\sigma_k^g + 2}, \tag{21}$$

$$g = \sqrt{|S| + |A|}, \tag{22}$$

$$G(k) = V(k) + B(k), \tag{23}$$

where Q_t represents the original population, P_t represents the archive population,l and k are the individuals in Q_t and P_t, μ_k^g grepresents the distance between the k-th individual to the g-th individual.

Then there are two selections: the environmental selection and the mating selection. The environmental selection: when a new population is needed to be constructed, the environmental selection should be made first. All the nondominated individuals which fitness is lower than one is selected from Q_t and P_t to the next archive population P_{t+1}, as calculated by:

$$P_{t+1} = \{k | k \in Q_t + P_t \wedge G(k) < 1\}. \tag{24}$$

If the size of the nondominated individuals is smaller than A, those excellent individuals with small fitness are selected from Q_t and P_t to P_{t+1} again. If the size of the nondominated individuals is larger than A, the archive truncation procedure is used to remove individuals in P_{t+1}, as calculated by:

$$\begin{aligned} \forall 0 < g < |P_{t+1}| : \\ \mu_k^g = \mu_l^g \vee \exists 0 < g < |P_{t+1}| : \\ [(\forall 0 < q < g : \mu_k^q = \mu_l^q) \wedge \mu_k^g = \mu_l^g], \end{aligned} \tag{25}$$

where μ_k^g represents the distance between the k-th individual and the g-th individual in the P_{t+1}.

The mating selection: the tournament selection is implemented in the P_{t+1} to make the next crossover and mutation operation.

4.6 Crossover and Mutation

The traditional single-point crossover operation is taken to combine the two parental chromosomes to generate two new chromosomes. In the crossover operation, the only one crossover point should be selected first, and the chromosomes should be changed next. Fig. 6 shows an example of crossover operation.

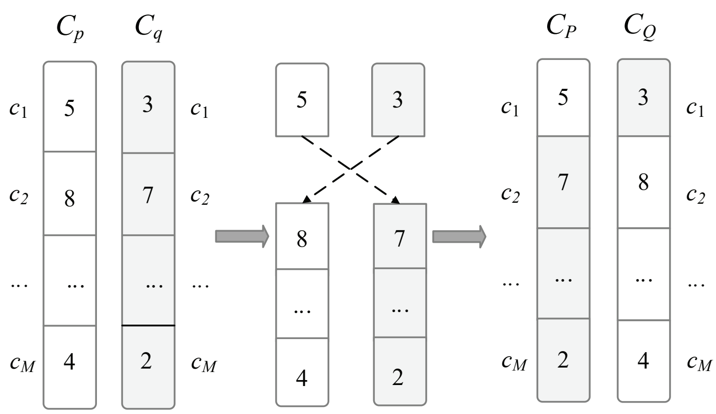


Fig. 6: An example of crossover operation.

When the offspring chromosomes are no longer better than their parental chromosomes while they do not reach the globally optimal solution, the premature convergence will happen. The mutation operation is taken to maintain individual diversity in the population. The probability of each gene which will mutate is equal. Fig. 7 shows an example of mutation operation.

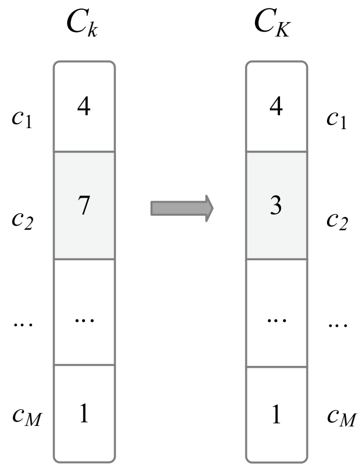


Fig. 7: An example of mutation operation.

4.7 Optimal Strategy Obtaining based on the TOPSIS and MCDM

The relative merits of the existing objects are sorted according to the distance between the limited number of evaluated objects and the ideal objective based on the method of TOPSIS [43]. MCDM is an efficient method which has been used in most of the multi-objective problems [44,45,46,47,48]. Both of the TOPSIS and the MCDM are used in this efficient computation offloading problem in this paper.

When various generated strategies are needed to be selected, the TOPSIS and the MCDM are applied to select the most efficient strategy. We propose that there are A strategies to be chosen, and there are three objectives for every strategy: the total offloading delay D_a, the total transmission cost C_a and the load balance degree L_a. All of the D_A, C_A and L_A consist of the three objectives. The steps which are made up by the method of TOPSIS and MCDM are as follows.

(1) The normalized value of offloading delay vd_a^D is calculated as:

$$vd_a^D = \frac{T_a}{\sqrt{\sum_{a=1}^{A} D_a^2}}, a = 1, ..., A. \tag{26}$$

We can set the normalized value of the transmission cost vd_a^C and the normalized value of the load balance degree vd_a^L like (24).

(2) The weight of the offloading delay, the transmission cost and the load balance degree can be denoted as wd_D, wd_C and wd_L, and $wd_D + wd_C + wd_L = 1$.

The weight of the normalized value of the offloading delay degree nd_a^D can be calculated as follows:

$$nd_a^D = wd_D \cdot vd_a^D, a = 1, 2, ..., A. \tag{27}$$

We normalize the transmission cost nd_a^C and the load balance degree nd_a^L like the offloading delay degree (25).

(3) In this problem to be solved, the load balance degree is the ideal solution, while the offloading delay and the transmission cost are the negative-ideal solutions. The maximum weight of the normalized value of the offloading delay, the transmission cost and the load balance degree are denoted as nd_{max}^D, nd_{max}^C and nd_{max}^L. The minimum weight of the normalized value of the offloading delay, the transmission cost and the load balance degree are denoted as nd_{min}^D, nd_{min}^C and nd_{min}^L.

(4) The distance between the alternative solution and the ideal solution can be calculated as follows:

$$\begin{aligned}(AI_a^*)^2 = (nd_a^D - nd_{min}^D)^2 + (nd_a^C - nd_{min}^C)^2 + \\ (nd_a^L - nd_{\max}^L)^2, a = 1, 2, ..., A.\end{aligned} \tag{28}$$

The distance between the alternative solution and the negative-ideal solutions can be calculated as follows:

$$\begin{aligned}(AI_a^-)^2 = (nd_a^D - nd_{max}^D)^2 + (nd_a^C - nd_{max}^C)^2 + \\ (nd_a^L - nd_{\min}^L)^2, a = 1, 2, ..., A.\end{aligned} \tag{29}$$

(5) The relative closeness of every alternative solution to the ideal solution can be calculated as follows:

$$SI_a^* = \frac{AI_a^-}{AI_a^- + AI_a^*}, a = 1, 2, ..., A. \tag{30}$$

(6) All of the alternative solutions are ranked by the sequence of the relative closeness. The process of making decision is as follows:

$$max SI_a^*, a = 1, 2, ..., A, \tag{31}$$

$$s.\ t. \quad wd_D, wd_C, wd_L \in [0, 1], \tag{32}$$

$$wd_D + wd_C + wd_L = 1. \tag{33}$$

In our method, we calculate utility value of the strategies generated by SPEA2, and the TOPSIS as well as MCDM are employed to pick out the most optimal strategy.

4.8 Method Review

In this paper, we aim at minimizing the offloading delay and the offloading cost and optimizing the load balance at the same time. The efficient computation offloading problem is regarded as a multi-objective problem, and the SPEA2 algorithm is selected to solve this problem because of its high performance in multi-objective optimization problem. First, the RSUs are encoded, and the fitness functions and the constraints are given for the efficient computation offloading problems. Then better chromosomes are selected from the population after the fine-grained fitness assignment strategy, the environmental selection and the mating selection. What's more, the crossover operation and the mutation operation are taken to avoid the premature convergence and generate new better offspring chromosomes. In GA, the number of individuals with good fitness is still large, so the TOPSIS and MCDM are used to select the optimal strategy. The following Fic.8 is the flowchart of SPEA2 algorithm.

Algorithm 5 Method overview of obtaining the optimal strategy

Require: Q_0
Ensure: P_B
1: **for** $i = 1$ to T **do**
2: Calculate the fitness of the individual in Q_t and P_t by formulas (19) - (23)
3: Make the environmental selection in Q_t and P_t by formulas (24) (25)
4: Make tournament selection in P_{t+1}
5: Make crossover and mutation operations
6: **end for**
7: Evaluate the individuals in P_T using TOPSIS and MCDM
8: Obtain the optimal strategy
9: **return** P_B

The overview of the proposed method is elaborated in Algorithm 2. We input the initialized population Q_1 and output the archive set of the maximum iteration P_T. In each iteration, we calculate the fitness of the individual in Q_t and P_t by (17) (18) (20) (21). Then we make the environment selection according to (22) (23) and produce P_{t+1}. What's more, for P_{t+1}, we do the tournament selection and finally we make crossover and mutation selection. In the Algorithm 5, we introduce the procedure of obtaining the optimal strategy, and Q_0 represents the initial population and the P_B represents the best strategy.

5 Experimental Evaluation

In this section, in order to evaluate the performance of the proposed COV method, a series of simulations and experiments are carried out. Firstly, the settings of the experimental environment are described in detail, including the simulation settings and the description of the comparison methods. Then, the most balanced strategies of COV at different vehicle scales are selected out by TOPSIS and MCDM. Lastly, the effects of the various vehicle scales on the performance of the offloading delay, the offloading cost and the loading balancing degree performed by the computation offloading methods are evaluated and compared.

5.1 Simulation Setup

In our experiments, we engage three datasets of vehicle scales, and the number of vehicles is set to 20, 40, 60, 80, 100 and 120. According to [49], the data transmission rate between the RSU and the vehicle is set to 3Mbps. Similarly, the data transmission rate between two ENs is set to 1.2 Gbps, and the data transmission rate between the RSU and the accessible EN is set to 1 Gbps according to [50] respectively. The total number of ENs is set to 20 and the total number of VMs in each EN is set to 20 as well. In this simulation, there are some VMs which are employed already by some other vehicle applications and the experiments are undergoing with those existed vehicle applications into consideration. The parameters and the corresponding values are specified in Table 2.

In order to conduct the comparison analysis, three computation offloading methods are adopted besides COV. The comparison method is summarized as follows.

- Benchmark: In this method, the vehicle applications are offloaded to the nearest RSU and transfer to the nearest EN for computing. However, when the vehicle application to be offloaded requires more computing resources than the nearest EN owns, this vehicle application is offloaded to the ENs near the current one according to Dijkstra's algorithm. Then this process is repeated until all vehicle applications are offloaded to the ENs.
- First fit decreasing-based computation offloading on resource-utilization optimization (FFD-RU): In this method, the vehicle applications are sorted decreasingly according to their requirement of the computing resource. Then the vehicle applications are offloaded to the ENs with enough computing resource according to the sorting. This process is repeated until all vehicle applications have been offloaded.

Table 2: Parameter Settings

Parameter	Value
The engaged datasets of the vehicle scales M	$\{20, 40, 60, 80, 100, 120\}$
The total number of ENs N	20
The number of VMs in each EN θ	20
The data transmission rate between the RSU and the vehicle tr_{VR}	3Mbps
The data transmission rate between two ENs tr_{EN}	1.2 Gbps
The data transmission rate between the RSU and the accessible EN tr_{RE}	1 Gbps
The number of the employed VMs for each va_m	[0,20]
The occupied memory size of each vehicle applications s_m	[5,9]

– Best fit decreasing-based computation offloading on resource-utilization optimization (BFD-RU): Under the condition of this method, the vehicle applications are sorted decreasingly by the resource they required before the computation offloading. Then the vehicle applications are offloaded to the ENs which owns the least but enough computing resource for the current application one by one. When all the vehicle applications have been offloaded, this process stops repeating.

Benchmark, FFD-RU and BFD-RU are the traditional approaches to solve the multi-objective optimization problem. The comparation amongst the proposed approach COV, benchmark, FFD and BFD are capable of showing the superiority of COV easily.

By employing the CloudSim simulation tool, the computational offloading methods are implemented on a desktop PC with Inter Core i7-8700 3.20 GHz processors and 16 GB RAM. The corresponding assessment results will be described in detail in the following subsections.

5.2 Strategy selection of COV

According to Section 4, as the COV generates couples of strategies, the TOPSIS and MCDM are employed to select the most optimal strategy in our experiments. Fig. 9 shows the utility value of the strategies generated by COV by different vehicle scales. It shows that when the number of vehicles is 20, 40, 60, 80, 100 and 120, the number of the strategies generated by COV is 3, 2, 3, 4, 3 and 3 respectively. After statistics and analysis, the strategy with the maximum utility value is employed as the most optimal strategy. From Fig. 9, the most balance strategies are solution 3, 1, 3, 1, 1 and 3 respectively.

5.3 Comparison analysis

In this subsection, the performances of Benchmark, FFD-RU, BFD-RU and COV are evaluated and compared in detail. The offloading delay, the offloading cost and the load balance degree are assumed as the main metrics to evaluate the performance of the proposed computation offloading methods. Moreover, the number of the employed ENs is presented to show the resource utilization of all the ENs for offloading the vehicle applications. The evaluation results are shown in the figures 2, 3, 4, 5 and 6 respectively.

1) Evaluation on the offloading delay: According to Section 3, the total offloading delay is composed by three different parts which are the data transmission between the vehicle and the RSU, the data transmission between the accessible EN and the RSU as well as the data transmission between two ENs. Fig. 10 illustrates the comparison of those three different parts of the offloading delay by the computation offloading methods. From Fig. 10 (a) and (b), the offloading delay between the vehicle and the RSU as well as the accessible EN and RSU keep the same on the condition of the same vehicle scale by FFD-RU, BFD-RU and COV. In addition, the two parts of offloading delay are increasing along with the increase of the vehicle scales. From Fig. 10 (c), the offloading delay of Benchmark is 0 because there is no offload taking place between ENs in Benchmark. In addition, when the number of vehicles is below 80, the offloading delay between two ENs raised by COV is a little more than FFD-RU and BFD-RU. However, with the increase of vehicle scales, FFD-RU and BFD-RU cost more offloading delay than COV and the gap is enlarging in this tendency. Fig. 11 shows the total offloading delay by four computation offloading methods. The differences of the offloading delay in the same vehicle scales performed by those computation offloading methods is resulted totally by the difference of the offloading delay between two ENs.

2) Evaluation on the offloading cost: In this paper, the offloading cost mainly refers to the transmission amount of the vehicle applications for the ENs because of the receiving and transferring the vehicle applications. More offloading times during ENs with more vehicle applications contributes to more offloading cost. Fig. 12 shows the comparison of the offloading cost performed by Benchmark, FFD-RU, BFD-RU and COV. It is intuitive that the Benchmark costs less than the other computation offloading methods, which is because there does not exist any transferring from ENs in Benchmark. Despite the Benchmark, when the number of vehicles is below 80, the offloading cost consumed by COV is little more than FFD-RU and BFD-RU. This is because when there are little vehicle applications, COV needs more offloading times between two ENs to improve the load balancing degree, which raises the offloading cost to some extent. On the other hand, when the number of vehicle applications increases, the offloading cost of COV is less than FFD-RU and BFD-RU. With the increase of the vehicle scales, the gap of the offloading during COV and FFD-RU as well as BFD-RU is enlarging, which means our proposed COV may more suitable when the vehicle applications are more than 100.

Table 3: Improvement of Load Balance Degree with COV Compared to Benchmark, FFD-RU and BFD-RU

Computation offloading methods	The number of vehicle applications					
	20	40	60	80	100	120
Benchmark	782.2%	492.8%	810.0%	3807.9%	3561.8%	2606.5%
FFD-RU	1057.1%	539.0%	520.0%	522.3%	2766.5%	1698.2%
BFD-RU	1057.16%	539.0%	520.0%	522.3%	2766.5%	1698.2%

3) Evaluation on the load balance degree: The load balance degree is calculated by the average utilization of the leveraged ENs and the active VMs in each EN, the smaller number of employed ENs and higher load balance degree means the computation offloading method is better resource utilized and load balanced. In Fig. 13, we compare the number of employed ENs to analyze the resource utilization of the computation offloading methods. It can be indicated from Fig. 13 that COV employed less ENs than Benchmark, FFD-RU and BFD-RU, which means COV is better resource utilized and wastes fewer computation resources in each EN. Then, the load balance degree of Benchmark, FFD-RU, BFD-RU and COV are compared in Fig. 14. In our experiment, some computing applications besides our vehicle applications have engaged some VMs in the ENs and those existed computing applications are taken into consideration along with the vehicle application to realize the load balance in our simulation. According to Section 3.3 a higher load balance degree means the computation offloading methods is more load balanced. It can be intuitive from Fig. 14 and Table 3 that our proposed COV has much better performance in load balancing than Benchmark, FFD-RU and BFD-RU. For instance, when the vehicle scale is 100, the improvements of COV to other methods are all near 3000%. This means that COV does well in the fair degree for the load distribution of the offloaded vehicle applications and reduce the phenomena raised by the uneven load distribution to a large extent.

6 Conclusion

In recent years, IoV has emerged as a powerful technology to transmit real-time traffic information to drivers. For fleet data processing rate, 5G networks are employed in IoV environment, represented by using base stations to receive and process offloaded vehicle applications. However, with the increase of vehicles, computing applications in vehicles become so complicated that it is of low efficiency to use macro base stations to process offloaded applications. In terms of pushing the computing services to the edge of the networks, edge computing paradigm is suitable to execute IoV computing applications in 5G networks. As the applications of the vehicles are offloaded to ENs in close proximity to vehicles, partial ENs may be overload and it is necessary to offload the applications in overload ENs to other idle ENs. Therefore, a computation offloading method for IoV named COV is proposed to realize the jointly optimization to reduce the application offloading delay and offloading cost across

ENs while achieving the load balance of ENs globally. First, edge computing framework in IoV is considered and described under the architecture of 5G networks. Then, SPEA2 is adopted to realize the multi-objective optimization to select appropriate destination ENs. Subsequent simulation experiments are conducted to demonstrate the effectiveness of COV.

7 Future Work

In future work, we will attempt to use other infrastructure as ENs and extend the proposed method to the real scenario in IoV environment. Furthermore, we will develop an offloading strategy to reduce application offloading delay and cost as well as achieving load balance of ENs within a set time.

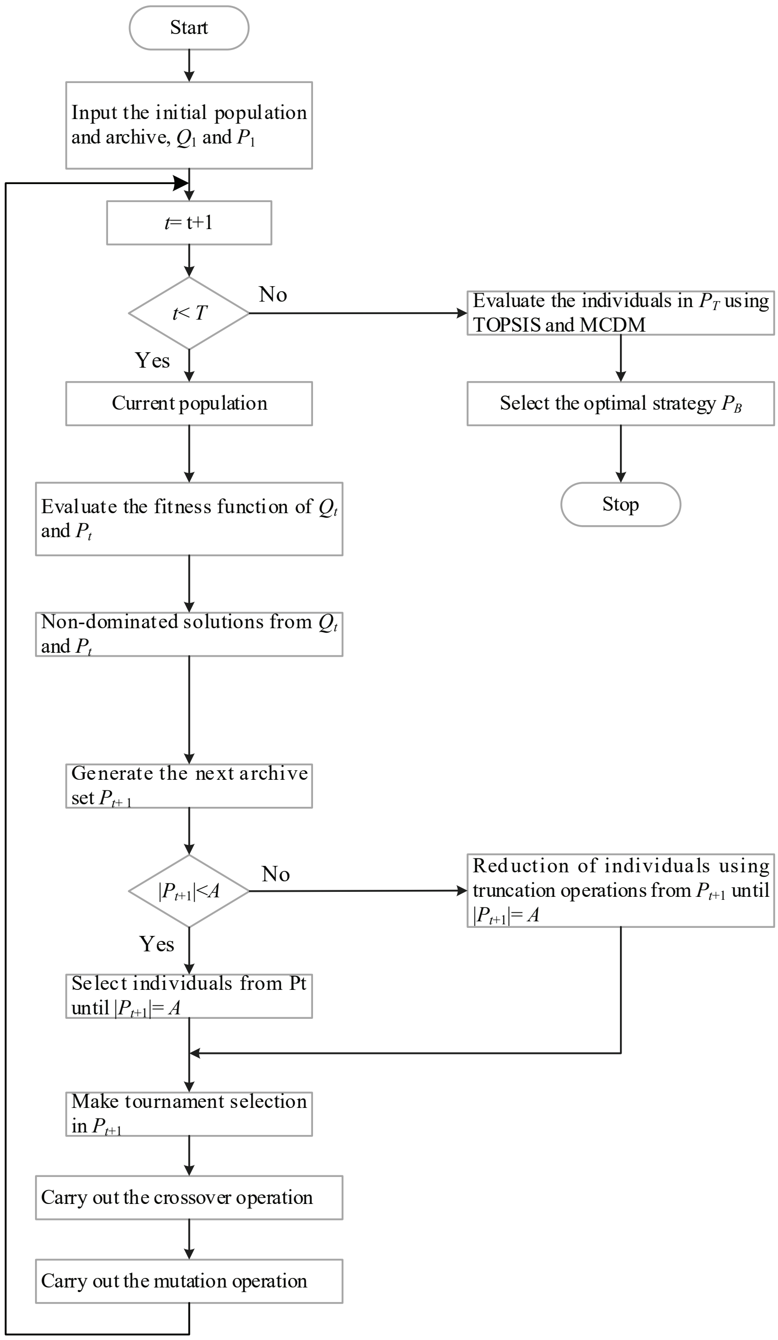
Acknowledgements This research is supported by the National Natural Science Foundation of China under grant no.61702277.

References

1. Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, M. S. Obaidat, A cooperative quality-aware service access system for social internet of vehicles, IEEE Internet of Things Journal 5 (4) (2018) 2506-2517.
2. K. M. Alam, M. Saini, A. E. Saddik, Toward social internet of vehicles: Concept, architecture, and applications, IEEE Access 3 (2015) 343-357.
3. J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, C. Liu, Routing in internet of vehicles: A review, IEEE Transactions on Intelligent Transportation Systems 16 (5) (2015) 2339-2352.
4. O. Kaiwartya, A. H. Abdullah, Y. Cao, M. Prasad, C. T. Lin, X. Liu, A. Altameem, Internet of vehicles: Motivation, layered architecture network model challenges and future aspects, IEEE Access 4 (2016) 5356-5373.
5. Y. Zhang, Y. Xiang, L. Y. Zhang, Y. Rong, S. Guo, Secure wireless communications based on compressive sensing: A survey, IEEE Communications Surveys & Tutorials (2018).
6. J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things, IEEE Internet of Things journal 1 (2) (2014) 112-121.
7. J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things, IEEE Internet of Things journal 1 (2) (2014) 112-121.
8. S. S. Belavadi, V. Malik, T. Udayakumar, A. Srinivas, R. Mohan, Iov based dynamic batch formation and scheduling technique for driverless vehicles, IEEE Region 10 Symposium (TENSYMP) (2017) 1-6.
9. M. Giordani, M. Mezzavilla, M. Zorzi, Initial access in 5g mm-wave cellular networks, IEEE Communications Magazine 54 (11) (2016) 40-47.
10. M. Giordani, M. Mezzavilla, N. Barati, S. Rangan, M. Zorzi, Comparative analysis of initial access techniques in 5g mmwave cellular networks, Conference on Information Science and Systems (2016) 268-273.
11. P. L. S. G. Y. S. J Mi, K Wang, Software-defined green 5g system for big data, IEEE Communications Magazine 56 (11) (2018) 116-123.
12. M. Giordani, M. Mezzavilla, S. Rangan, M. Zorzi, An efficient uplink multi-connectivity scheme for 5g mmwave control plane applications, IEEE Transactions on Wireless Communications 17(10) (2018) 6806-6821.
13. D. Zeng, J. Zhang, L. Gu, S. Guo, J. Luo, Energy-efficient coordinated multipoint scheduling in green cloud radio access network, IEEE Transactions on Vehicular Technology 67 (10) (2018) 9922-9930.

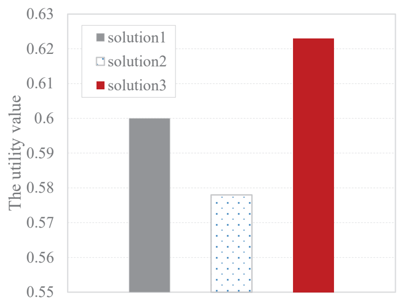
14. C. Liu, M. Li, S. V. Hanly, P. Whiting, I. B. Collings, Millimeter-wave small cells: Base station discovery, beam alignment, and system design challenges, IEEE Wireless Communications 25 (4) (2018) 40-46.
15. F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, P. Popovski, Five disruptive technology directions for 5g, IEEE Communications Magazine 52 (2) (2014) 74-80.
16. J. Feng, Z. Liu, C. Wu, Y. Ji, Ave: Autonomous vehicular edge computing framework with aco-based scheduling, IEEE Transactions on Vehicular Technology 66(12) (2017) 10660-10675.
17. K. Zhang, Y. Mao, S. Leng, Y. He, Y. Zhang, Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading, IEEE Vehicular Technology Magazine 12 (2) (2017) 36-44.
18. Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile edge computing—a key technology towards 5g, ETSI white paper 11 (11) (2015) 1-16.
19. T. X. Tran, A. Hajisami, P. Pandey, D. Pompili, Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges, IEEE Communications Magazine 55 (4) (2017) 54-61.
20. M. H. Eiza, Q. Ni, Q. Shi, Secure and privacy-aware cloud-assisted video reporting service in 5g-enabled vehicular networks, IEEE Transactions on Vehicular Technology 65 (10) (2016) 7868-7881.
21. M. E. Khoda, M. A. Razzaque, A. Almogren, M. M. Hassan, A. Alamri, A. Alelaiwi, Efficient computation offloading decision in mobile cloud computing over 5g network, Mobile Networks and Applications 21 (5) (2016) 777-792.
22. T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1657-1681.
23. S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, A. Schneider, Enabling real-time context-aware collaboration through 5g and mobile edge computing, 2015 12th international conference on information technology-new generations (ITNG), IEEE (2015) 601-605.
24. S. D. T. Kelly, N. K. Suryadevara, S. C. Mukhopadhyay, Towards the implementation of iot for environmental condition monitoring in homes, IEEE Sensors Journal 13 (10) (2013) 3846-3853.
25. S. Sarkar, S. Chatterjee, S. Misra, Assessment of the suitability of fog computing in the context of internet of things, IEEE Transactions on Cloud Computing 6 (1) (2018) 46-59.
26. X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, W. Dou, An Energy-Aware Computation Offloading Method for Smart Edge Computing in Wireless Metropolitan Area Networks, Journal of Network and Computer Applications, 133 (2019) 75-85.
27. S. K. Datta, J. Haerri, C. Bonnet, R. F. Da Costa, Vehicles as connected resources: Opportunities and challenges for the future, IEEE Vehicular Technology Magazine 12 (2) (2017) 26-35.
28. Y. Cao, Y. Chen, Qoe-based node selection strategy for edge computing enabled internet-of-vehicles (ec-iov), in: Visual Communications and Image Processing (VCIP) IEEE, (2017) 1-4.
29. X. Xu, Y. Xue, Y. Yuan, L. Qi, X. Zhang, T. Umer, S. Wan, An Edge Computing-Enabled Computation Offloading Method with Privacy Preservation for Internet of Connected Vehicles. Future Generation Computer Systems 96 (2019) 89-100.
30. K.Wang, H. Yin, W. Quan, G. Min, Enabling collaborative edge computing for software-defined vehicular networks, IEEE Network 99 (2018) 1-6.
31. Z.Wu, Z. Lu, P. C. Hung, S.-C. Huang, Y. Tong, Z.Wang, Qamec: A qos-driven iovs application optimizing deployment scheme in multimedia edge clouds, Future Generation Computer Systems 92 (2019) 17-28.
32. X. Xu, S. Fu, Y. Yuan, Y. Luo, L. Qi, W. Lin, W. Dou, Multi-Objective Computation Offloading for Workflow Management in Cloudlet-Based Mobile Cloud Using NSGA-II, Computational Intelligence (2018).
33. A. C. Baktir, A. Ozgovde, C. Ersoy, How can edge computing bene
t from software-defined networking: a survey, use cases, and future directions, IEEE Communications Surveys & Tutorials 19 (4) (2017) 2359-2391.

34. N. Kumar, S. Zeadally, J. J. Rodrigues, Vehicular delay-tolerant networks for smart grid data management using mobile edge computing, IEEE Communications Magazine 54 (10) (2016) 60-66.
35. Zhu. J, Chan. D. S, Prabhu. M. S, Natarajan. P, Hu. H, Bonomi. F, Improving web sites performance using edge servers in fog computing architecture, In 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering (2013) 320-323.
36. K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks, IEEE access 4 (2016) 5896-5907.
37. X. Chen, L. Pu, L. Gao, W. Wu, D. Wu, Exploiting massive d2d collaboration for energy-efficient mobile edge computing, IEEE Wireless Communications 24 (4) (2017) 64-71.
38. J. Guo, H. Zhang, L. Yang, H. Ji, X. Li, Decentralized computation offloading in mobile edge computing empowered small-cell networks, GlobecomWorkshops (GC Wkshps) IEEE (2017) 1-6.
39. L. Yang, H. Zhang, M. Li, J. Guo, H. Ji, Mobile edge computing empowered energy efficient task offloading in 5g, IEEE Transactions on Vehicular Technology 67(7) (2018) 6398-6409.
40. Bastug. E, Bennis. M, Debbah. M, Living on the edge: The role of proactive caching in 5G wireless networks. IEEE Communications Magazine, 52(8) (2014) 82-89.
41. M. Agiwal, A. Roy, N. Saxena, Next generation 5g wireless networks: A comprehensive survey, IEEE Communications Surveys & Tutorials 18 (3) (2016) 1617-1655.
42. X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, L. Qi, A Computation Offloading Method over Big Data for IoT-Enabled Cloud-Edge Computing, Future Generation Computer Systems 95 (2019) 522-533.
43. Siahaan. A. P. U, Mesran. M, Rahim. R, Technical Approach of TOPSIS in Decision Making, (2017).
44. Wang. P, Zhu. Z, Wang. Y, A novel hybrid MCDM model combining the SAW, TOPSIS and GRA methods based on experimental design, Information Sciences 345 (2016), 27-45.
45. Wan S., Zhao, Y., Wang, T., Gu, Z., Abbasi, Q. H., & Choo, K. K. R. Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. Future Generation Computer Systems, 91, 382-391, 2019.
46. Gao, Z., Wang, D. Y., Wan, S. H., Zhang, H., & Wang, Y. L. Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval. Future Generation Computer Systems, 94, 641-653, 2019.
47. Gao, Z., Xuan, H.Z., Zhang, H., Wan, S. and Choo, K.K.R. Adaptive fusion and category-level dictionary learning model for multi-view human action recognition. IEEE Internet of Things Journal, 2019.
48. Liu, J., Wang, W., Li, D., Wan, S., & Liu, H. Role of Gifts in Decision Making: An Endowment Effect Incentive Mechanism for Offloading in the IoV. IEEE Internet of Things Journal, 2019.
49. X. Li, B.-J. Hu, H. Chen, G. Andrieux, Y. Wang, Z.-H. Wei, An rsu-coordinated synchronous multi-channel mac scheme for vehicular ad hoc networks, IEEE Access 3 (2015) 2794-2802.
50. X. Ge, S. Tu, G. Mao, C.-X. Wang, T. Han, 5g ultra-dense cellular networks, IEEE Wireless Communications 23 (1) (2016) 72-79.

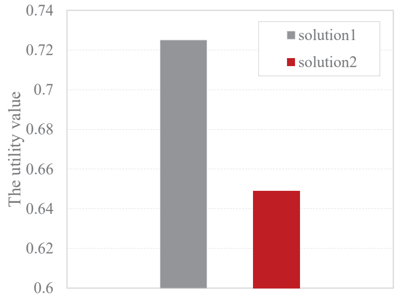


]

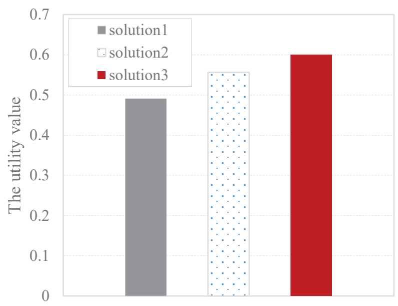
Fig. 8: the flow chart of selecting the optimal strategy.



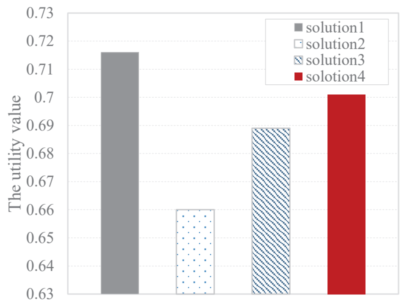
(a) The vehicle scale is 20



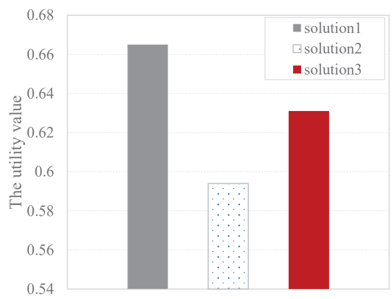
(b) The vehicle scale is 40



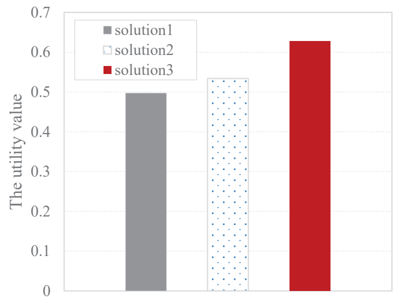
(c) The vehicle scale is 60



(d) The vehicle scale is 80

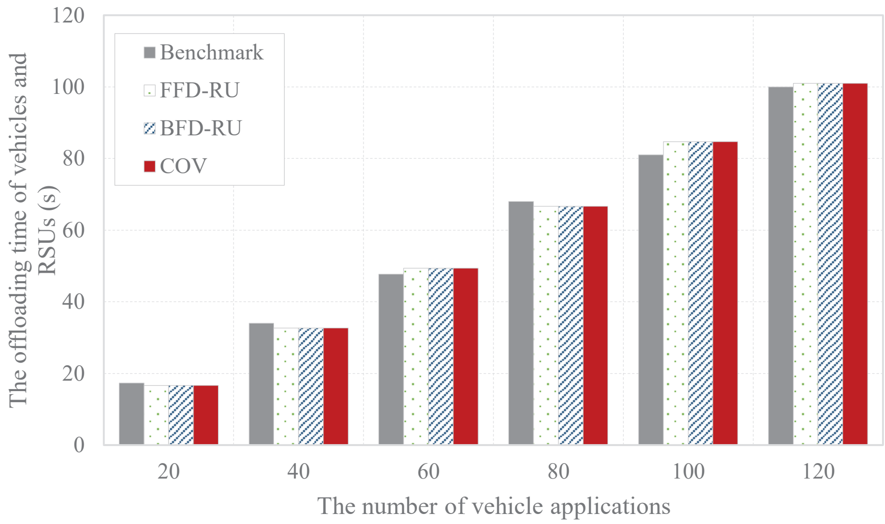


(e) The vehicle scale is 100

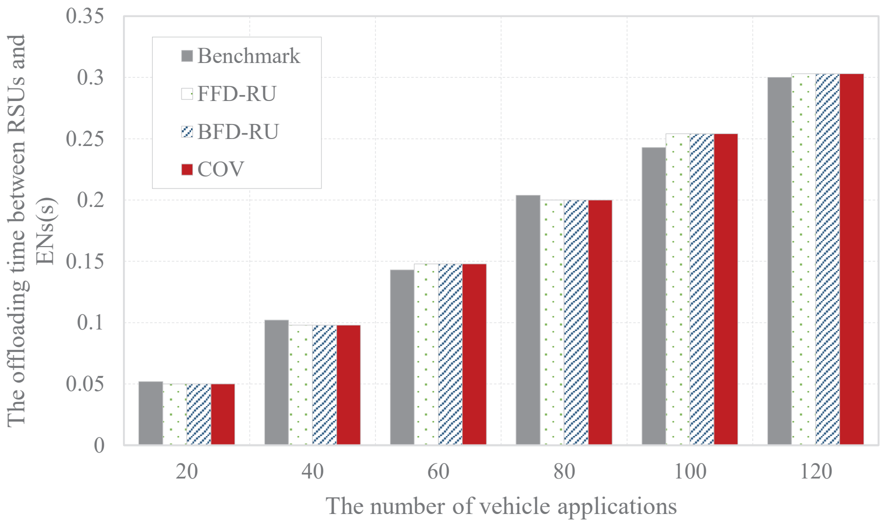


(f) The vehicle scale is 120

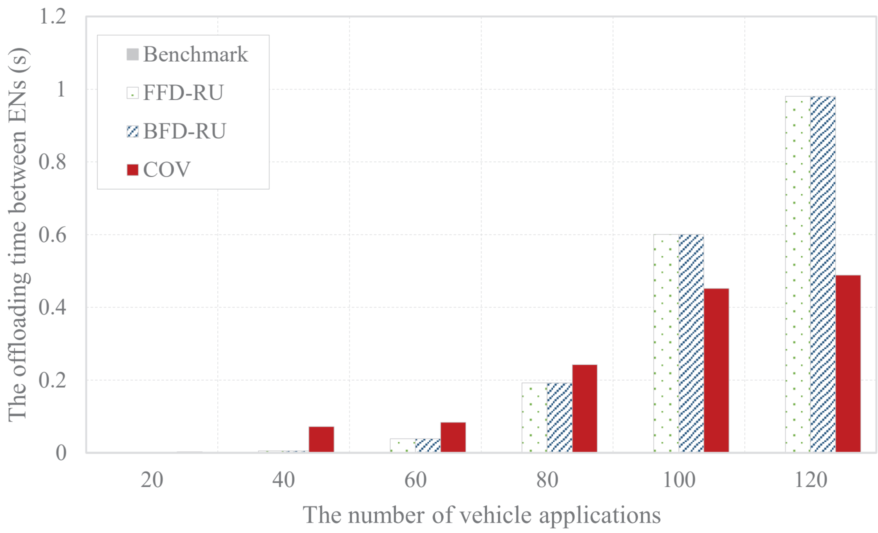
Fig. 9: The utility value performed by TOPSIS and MCDM of COV at different vehicle scales.



(a) Offloading delay between vehicles and RSUs



(b) Offloading delay between ENs and RSUs



(c) Offloading delay between ENs

Fig. 10: Comparison of the different parts in the offloading delay by Benchmark, FFD-RU, BFD-RU and COV at different vehicle scales.

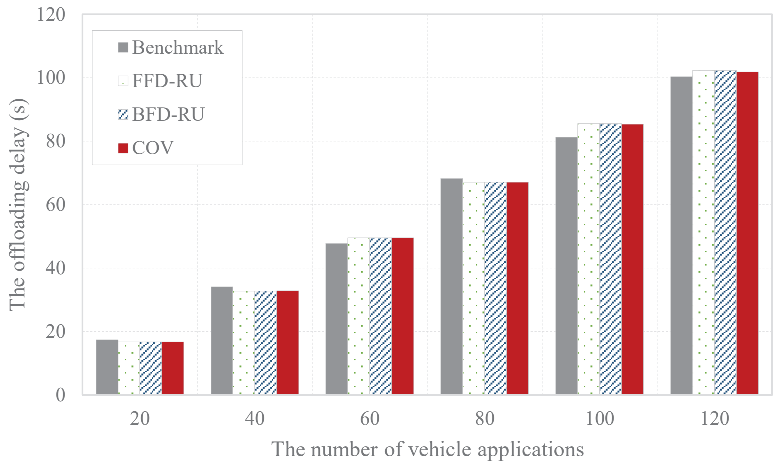


Fig. 11: Comparison of the offloading delay by Benchmark, FFD-RU, BFD-RU and COV at different vehicle scales.

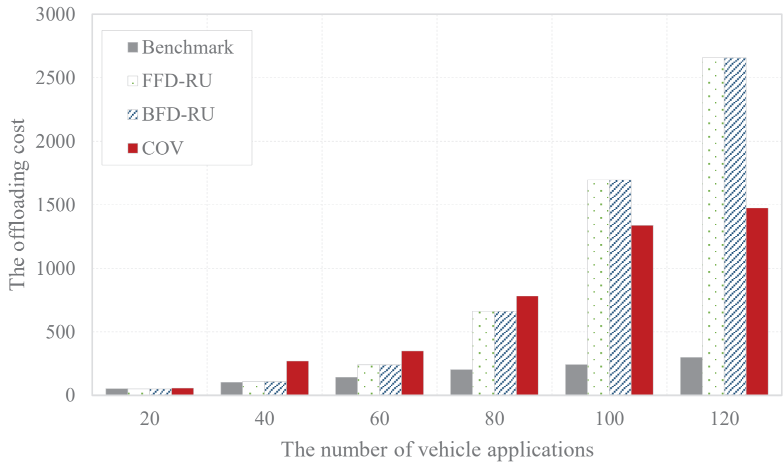


Fig. 12: Comparison of the offloading cost by Benchmark, FFD-RU, BFD-RU and COV at different vehicle scales.

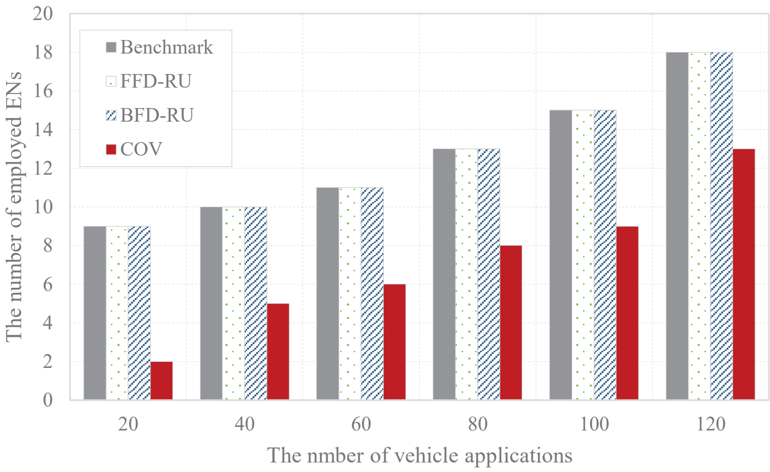


Fig. 13: Comparison of the number of employed ENs by Benchmark, FFD-RU, BFD-RU and COV at different vehicle scales.

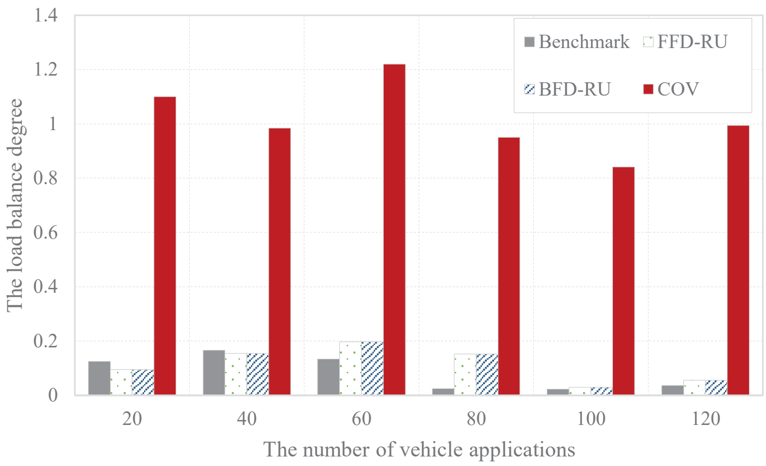


Fig. 14: Comparison of the load balance degree by Benchmark, FFD-RU, BFD-RU and COV at different vehicle scales.