Accepted Manuscript

An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles

Xiaolong Xu, Yuan Xue, Lianyong Qi, Yuan Yuan, Xuyun Zhang, Tariq Umer, Shaohua Wan

PII: S0167-739X(18)31870-3
DOI: https://doi.org/10.1016/j.future.2019.01.012
Reference: FUTURE 4705

To appear in: *Future Generation Computer Systems*

Received date : 2 August 2018
Revised date : 7 December 2018
Accepted date : 9 January 2019

Please cite this article as: X. Xu, Y. Xue, L. Qi et al., An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.01.012

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

An Edge Computing-Enabled Computation Offloading Method with Privacy Preservation for Internet of Connected Vehicles

Xiaolong Xu[a,b,c], Yuan Xue[a,b], Lianyong Qi[d], Yuan Yuan[e], Xuyun Zhang[f], Tariq Umer[g], Shaohua Wan[h,*]

[a] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[b] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China
[c] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[d] School of Information Science and Engineering, Qufu Normal University, China
[e] Department of Computer Science and Engineering, Michigan State University, MI, USA
[f] Department of Electrical and Computer Engineering, University of Auckland, New Zealand
[g] Department of Computer Science, COMSATS University Islamabad, Wah Campus, Pakistan
[h] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China

Abstract

The Internet of connected vehicles (IoV) is employed to collect real-time traffic conditions for transportation control systems, and the computing tasks are available to be offloaded from the vehicles to the edge computing devices (ECDs) for implementation. Despite numerous benefits of IoV and ECDs, the wireless communication for computation offloading increases the risk of privacy leakage, which may consequently lead to tracking, identity tampering and virtual vehicle hijacking. Therefore, it remains a challenge to avoid privacy conflicts for computation offloading to the ECDs in IoV. To address this challenge, an edge computing enabled computation offloading method, named ECO, with privacy preservation for IoV is proposed in this paper. Technically, the privacy conflicts of the computing tasks in IoV are analyzed in a formalized way. Then,

*Shaohua Wan is the corresponding author
Email addresses: njuxlxu@gmail.com (Xiaolong Xu), xueyuannuist@gmail.com (Yuan Xue), lianyongqi@gmail.com (Lianyong Qi), yyuan@msu.edu (Yuan Yuan), xuyun.zhang@auckland.ac.nz (Xuyun Zhang), t.umer@ciitwah.edu.pk (Tariq Umer), shaohua.wan@ieee.org (Shaohua Wan)

ACCEPTED MANUSCRIPT

vehicle-to-vehicle (V2V) communication-based routing for a vehicle is designed to acquire the routing vehicles from the origin vehicle where the computing task is located at the destination vehicle. NSGA-II (non-dominated sorting genetic algorithm II) is adopted to realize multi-objective optimization to reduce the execution time and energy consumption of ECDs and prevent privacy conflicts of the computing tasks. Finally, experimental evaluations are conducted to validate the efficiency and effectiveness of ECO.

Keywords: IoV, privacy preservation, edge computing, computation offloading, energy consumption

2010 MSC: 00-01, 99-00

---

1. Introduction

1.1. Background

In recent years, the number of vehicles has increased rapidly to expand residents' travel range, thus stretching transportation systems to their capacity limits [1]. With the explosive growth of vehicles, traffic congestion and car accidents occur frequently in urban areas. To improve traffic conditions in urban cities, the Internet of connected vehicles (IoV) has emerged as a new paradigm that emphasizes information interaction among vehicles and humans. In the IoV environment, vehicles are connected to devices such as intelligent cameras, sensors and actuators. These devices have transmitters and receivers that connect the vehicles to the remote infrastructure and other vehicles [2][3]. The real-time traffic information collected from the vehicles includes vehicle position, vehicle safety, vehicle driving status and vehicle identification information. IoV services can increase the dissemination of real-time traffic information and the ability of the vehicle drivers to track traffic conditions in real time.

Generally, most vehicles are not equipped with the physical resources for data processing and data storage, the computing tasks from the running vehicles currently need to be offloaded to the remote cloud data centers for implementation

ACCEPTED MANUSCRIPT

via roadside units (RSUs) based on the vehicle-to-infrastructure (V2I) communication mode [4]. Despite the on-demand processing ability and nearly endless resource supplement of cloud platforms, traditional centralized cloud computing has the drawback of excessive transmission delays and unstable connections in the IoV network [5][6]. Mobile edge computing (MEC) has the potential to provide a suitable solution for these issues. Driven by the MEC technique, cloud services are pushed to the edge of the radio access network and the computing tasks of the vehicles are offloaded to the edge computing devices (ECDs) that are in close proximity to the vehicles rather than remote cloud platforms [7]. By offloading the workloads to the ECDs, the quality of experience of the drivers in executing their computing tasks could be greatly improved.

However, the offloading process across ECDs in IoV comes with its own weaknesses, particularly in terms of security issues. The running vehicles are required to transmit security information, including driving speeds, current locations and surrounding traffic conditions, periodically to the transportation control center via their neighbor RSUs [8][9]. Once the privacy information is transmitted, the network could realize the whereabouts of the specific vehicle, which may bring the risk of privacy leakage [10][11]. The disclosure of private driving information is a catastrophe, especially in the aspect of intelligent driving. Although the use of sensors automates and intellectualizes the operations of vehicles and improves the safety for on-road traveling, it leads to the easy invasion of vehicles' systems. Virtual vehicle hijacking is likely to occur when the vehicles' electronic systems are invaded, resulting in unpredictable consequences. The schemers who invade the systems could alter the settings of the vehicles and give the systems extra instructions within a short time [12][13]. When the drivers are preoccupied with driving, untimely orders could be issued, such as disabling brakes and locking doors and windows, which remains a classic example in virtual vehicle hijacking. Besides, based on the security information online, the malicious drivers may trace the target vehicles and then carry out their criminal activities [14]. With these observations, as the computing tasks carry different privacy information of the vehicles, it is necessary to

ACCEPTED MANUSCRIPT

avoid privacy leakage during computation offloading among ECDs in IoV.

On the other hand, although ECDs are practical in IoV, due to the limited computing power of the ECDs, some of the computing tasks in the coverage of an ECD must be offloaded to the other ECDs for the resource response, which leads to a certain amount of extra communication delay by task transmission across ECDs [15]. Additionally, a large quantity of resources are deployed on the ECDs and the cloud data centers to cope with the explosive resource requirements of computing tasks in IoV. Therefore, from the perspective of ECDs, the energy consumption should be comprehensively considered. The placement of large quantities of ECDs consumes high amounts of energy, which is environmentally unfriendly. This ever-growing energy consumption will contribute to increased greenhouse gas emissions, worsening the greenhouse effect. Hence, it is of utmost significance to formulate relevant strategies, such as switching ECDs with low vehicle coverage to a sleep state so that non-participation will not influence global performance [16].

1.2. Motivation

To improve the response time for performing computing tasks for the vehicles in IoV, the ability of RSUs has been expanded as ECDs to provision computation and storage power for computing tasks. However, when employing ECDs to accommodate offloaded computing tasks, the resource limitations of ECDs should be prioritized. That is, the number of simultaneously running tasks on an ECD must be restricted. In this situation, the computing tasks in the coverage of an ECD may be transferred to another ECD for execution. For computation offloading across ECDs, the response times of all ECDs should be improved, and energy consumption by the servers in the ECDs should be reduced. Furthermore, as the datasets for running the computing tasks have privacy conflicts, some computing tasks cannot be offloaded to the same ECD to prevent privacy leakage.

1.3. Paper Contributions

The main contributions of this paper include the following:

ACCEPTED MANUSCRIPT

- Analyze privacy conflicts of the computing tasks offloaded to the ECDs, and the computation offloading problem with privacy preservation for IoV is defined as a standard multi-objective optimization problem.
- Design a V2V communication-based route-obtaining algorithm to acquire the vehicle route from the origin vehicle where the computing task is located to the destination vehicle that offloads computing tasks to the goal ECD.
- Adopt NSGA-II (non-dominated sorting genetic algorithm II) to realize multi-objective optimization to shorten the execution time of the computing tasks and reduce the energy consumption of ECDs while guarding against privacy conflicts of the computing tasks.
- Conduct extensive experimental evaluations to demonstrate the efficiency and effectiveness of the proposed method ECO.

The remainder of this paper is organized as follows. Section II describes the completed mathematical modeling and formulation. Section III develops a computation offloading method with privacy preservation for IoV in edge computing. In Section IV, simulation experiments and a comparison analysis are presented. Section V summarizes the related work. Finally, conclusions and future work are outlined in Section VI.

2. System Model and Problem Formulation

In this section, the system model for IoV in cloud-edge computing is designed and the computation offloading problem with privacy preservation for IoV is defined as a standard multi-objective optimization problem. Key terms and descriptions are presented in Table I.

2.1. Resource Model for IoV in Cloud-Edge Computing

The emerging paradigm of edge computing has the potential to satisfy the requirements of computation power for the computing tasks from vehicles in IoV.

ACCEPTED MANUSCRIPT

Table 1: Key Terms and Descriptions

Terms	Descriptions
M	The number of ECDs
D	The ECD collection, $D = \{d_1, d_2, \ldots, d_M\}$
R	The RSU collection, $R = \{r_1, r_2, \ldots, r_M\}$
S	The server collection, $S = \{s_1, s_2, \ldots, s_M\}$
N	The number of vehicles
V	The vehicle collection, $V = \{v_1, v_2, \ldots v_N\}$
q	The capacity of all servers
T	The computing task set, $T = \{t_1, t_2, \ldots, t_N\}$
t_n	The n-th computing task in T
u_n	The requested number of resource units of t_n
G	The time consumption for implementing T
BE	The baseline power consumption for all servers
RE	The energy consumed by the used resource units
UE	The energy consumed by the unused resource units
E	The total energy consumption for all servers

Fig. 1 shows a communication framework for IoV in cloud-edge computing. As indicated in Fig. 1, consider a scenario in which there is a bidirectional road and M edge computing devices (ECDs), denoted as $D = \{d_1, d_2, \ldots, d_M\}$, along the road, and N vehicles moving down the road, denoted as $V = \{v_1, v_2, \ldots v_N\}$. Suppose each vehicle has a computing task for offloading to the ECDs; thus, there are N computing tasks, denoted as $T = \{t_1, t_2, \ldots, t_N\}$. Each ECD consists of a roadside unit (RSU) and a server. Accordingly, there are M RSUs, denoted as $R = \{r_1, r_2, \ldots, r_M\}$, and M servers, denoted as $S = \{s_1, s_2, \ldots, s_M\}$, in the bidirectional road. The RSU often has a coverage area, and thus, the road is divided into multiple road segments. A vehicle in IoV transmits its data and applications to the corresponding surrounding ECD that covers the vehicle. The

ACCEPTED MANUSCRIPT

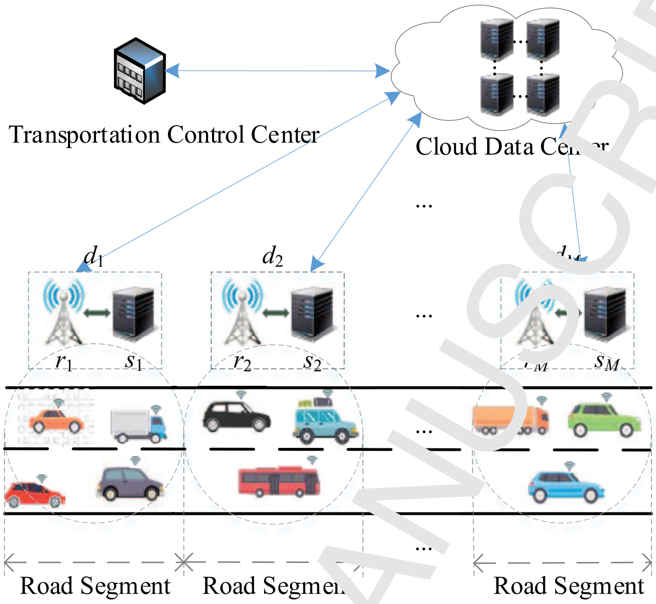


Figure 1: A communication framework for IoV in cloud-edge computing.

ECDs then transfer the collected information to the remote cloud data center. In addition, the transportation control center can reserve or retrieve the data from the cloud data center.

2.2. Execution Time Model

When performing a computing task from a vehicle, the offloading time from the vehicle to the ECD, the execution time and the feedback time for returning the execution results back to the vehicles should be considered.

The running vehicles cross different ECDs according to their locations along the road. We use a flag to judge whether the n-th ($n = \{1, 2, \ldots, N\}$) vehicle v_n belongs to the service domain of the m-th ECD at time instant i, which is measured by

$$F_n^m(i) = \begin{cases} 0, & v_n \text{is within the coverage of } d_m, \\ 1, & \text{Otherwise.} \end{cases} \tag{1}$$

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Generally, the computing task is chosen to be offloaded to an edge device. However, the computing task might be offloaded to an edge computing device that does not belong to the coverage of the nearby edge computing device. As the vehicle moves along the road, especially when it is located between the boundary of two edge devices, it may cross two or more road segments. Suppose the vehicle moves from the original segment to the destination segment. The task should be offloaded to the EDC located in the destination segment, where the vehicle will receive the feedback execution results. The computing task should be transmitted to the vehicle in the destination segment based on V2V technology.

The transmission time for transiting the computing task t_n is calculated by

$$b_n(i) = \sum_{m=1}^{M} \sum_{n'=1}^{N} F_n^m(i) \cdot Q_n^{n'}(i) \cdot (1 - F_{n'}^{m}(i)) \cdot \frac{w_n}{\lambda_{\mathrm{V2V}}} \cdot (\theta_{n,n'} + 1), \tag{2}$$

where $\theta_{n,n'}$ is the number of routing vehicles that transferred from v_n to $v_{n'}$, λ_{V2V} is the data transmission rate based on V2V technology, and $Q_n^{n'}(i)$ is a binary variable that judges whether t_n is transmitted from v_n to $v_{n'}$ and is calculated by

$$Q_n^{n'}(i) = \begin{cases} 0, & t_n \text{ is transmitted from } v_n \text{ to } v_{n'}, \\ 1, & \text{otherwise} \end{cases}. \tag{3}$$

The offloading time of the m-th ($m = \{1, 2, \ldots, M\}$) computing task t_m is calculated by

$$c_n(i) = \sum_{m=1}^{M} F_n^m(i) \cdot \frac{w_n}{\lambda_{\mathrm{V2I}}}, \tag{4}$$

where λ_{V2I} is the data transmission rate based on V2I technology. The execution time is determined by the execution performance of the resource units and the task length. In resource management, a common method is to employ resource units to measure the capacity of the server. Thus, the physical resources of the servers in the ECDs could be configured as multiple resource units, and then, the server capacity and the requested resources of the computing tasks could be weighted by the quantity of resource units. Let q be the capacity of all servers and u_n be the requested number of resource units of t_n.

ACCEPTED MANUSCRIPT

The execution time of t_n is calculated by

$$k_n(i) = \sum_{m=1}^{M} F_n^m(i) \cdot \frac{l_n}{u_n \cdot p}, \tag{5}$$

where p is the processing power of each resource unit.

The execution results should be fed back to the vehicles, and the feedback time is calculated by

$$h_n(i) = w'_n / \lambda_{\mathrm{V2I}}, \tag{6}$$

where w'_n is the data size of the returned results for executing t_n.

The total time consumption for implementing t_n is calculated by

$$g_n(i) = b_n(i) + c_n(i) + k_n(i) + h_n(i). \tag{7}$$

Then, the total time consumption for implementing all computing tasks is calculated by

$$G = \sum_{n=1}^{N} g_n(i). \tag{8}$$

2.3. Energy Consumption Model

The energy consumption of the ECDs mainly refers to the energy consumption of the servers and the energy consumed by the RSUs. As RSUs are in working mode, and their energy consumption is adjusted dynamically according to their working status. Thus, we mainly focus on the energy consumption of the servers in the ECDs. The main energy consumption of the servers includes several aspects: the baseline energy consumption of the servers in running mode, the energy consumption of the occupied resource units, and the energy consumption of the unused resource units [17].

The main energy consumption is determined by the service time of the servers. The service time of s_m is calculated by

$$st_m(i) = \max_{n=1}^{N} (L_n^m(i) \cdot k_n(i)), \tag{9}$$

where $L_n^m(i)$ is a binary variable that judges whether t_n is performed on s_m

$$L_n^m(i) = \begin{cases} 0, & t_n \text{ is executed on } s_m, \\ 1, & \text{otherwise} \quad .. \end{cases} \tag{10}$$

ACCEPTED MANUSCRIPT

The baseline energy consumption for all servers in the ECDs is calculated by

$$BE = \sum_{m=1}^{M} st_m(i) \cdot \alpha, \tag{11}$$

where α is the power rate of the servers in the ECDs.

The energy consumption of the employed resource units is calculated by

$$RE = \sum_{m=1}^{M} \sum_{n=1}^{N} L_n^m(i) \cdot st_m(i) \cdot \beta, \tag{12}$$

where β is the power rate of the employed resource units.

The energy consumption of the unemployed resource units is calculated by

$$UE = \sum_{m=1}^{M} (q - \sum_{n=1}^{N} L_n^m(i)) \cdot st_m(i) \cdot \gamma, \tag{13}$$

where γ is the power rate of the unemployed resource units.

Then, the total energy consumption of all servers is calculated by

$$E = BE + RE + UE. \tag{14}$$

2.4. Privacy Model of Computing Tasks

The computing tasks from the vehicles in IoV combine privacy conflicts. These tasks are from different vehicles and require different datasets to achieve their goals. The datasets may have different privacy preservation requirements. The privacy conflict occurs in the scenario that an ECD processes the tasks whose needed datasets should not be placed together. Provided that the tasks with privacy conflicts are transmitted to the same ECD, the privacy of drivers is invaded to a great extent. Thus, some computing tasks are incapable of deployment in the same EDC for execution.

A graph $G = (T, Z)$ is leveraged to model the privacy conflicts of the computing tasks, where T is the set of computing tasks and Z is the set of conflicting relations. A pair of conflict relations $(t_n, t_{n'})$ $(t_n, t_{n'} \in T)$ cannot be deployed on the same ECD to guarantee the privacy information of the vehicles. The conflict computing tasks for t_n can be acquired according to

$$ct_n = \{t_{n'} | (t_n, t_{n'}) \in Z, n' = \{1, 2, ..., N\}\}. \tag{15}$$

ACCEPTED MANUSCRIPT

Denote the computation offloading strategy for all computing tasks as $X = \{x_1, x_2, \ldots, x_N\}$ $(x_n \in D)$, where x_n represents the destination edge computing device for hosting t_n. Then, according to the obtained conflicting task set for hosting t_n, the deployed location x_n also has a conflicting ECD set, which is obtained by

$$cd_n = \{x_j | x_j \in ct_n, j = \{1, 2, ..., |ct_n|\}\} \tag{16}$$

2.5. Problem Formulation

In this paper, we aim to achieve the goal of minimizing the execution time presented in (7) and reducing the energy consumption presented in (13) while meeting the privacy constraints. The formalized problem is given as

$$\max G, \; \min E. \tag{17}$$

$$s.\ t. \quad x_n \in D, \tag{18}$$

$$\sum_{n=1}^{N} \theta_n \cdot f_{n,w} \leq p, \tag{19}$$

$$x_n \notin cd_n, \tag{20}$$

3. A Computation Offloading Method with Privacy Preservation for IoV in Edge Computing

In this section, we propose an algorithm to obtain the offloading route for the computing tasks based on the V2V transmission first. Then, NSGA-II is adopted to find the global optimal offloading strategy.

3.1. Routing Obtaining Based on V2V Transmission

V2V is employed for task transmission crossing of different ECDs by using the existing vehicular communications, including IEEE 802.11p-based dedicated short-range communications (DSRC), WiFi, Bluetooth, ZigBee, and fourth generation (4G) [18] [19][20]. Through V2V communication, the computing task

ACCEPTED MANUSCRIPT

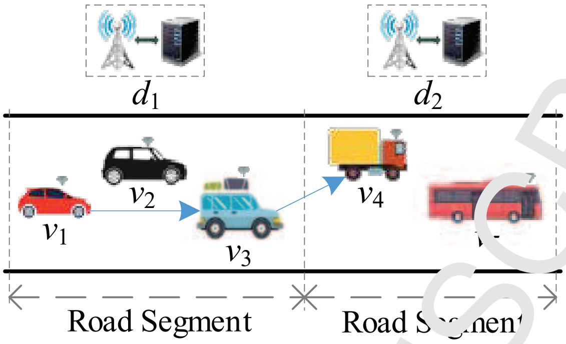


Figure 2: An example of route acquisition.

on the vehicle can be transmitted to another vehicle covered by the destination ECD and successfully offloaded to the ECD for execution.

As for the given task t_n from the vehicle v_n, the offloading goal ECD is x_n, and its origin ECD is d_n. The origin location of the ECD is denoted as $Ori_n = \{Olat_n, Olon_n\}$, and the destination location of the ECD is denoted as $Des_n = \{Dlat_n, Dlon_n\}$. The distance between the origin and the destination ECDs is calculated by

$$Dis_{OD} = \sqrt{(Olat_n - Dlat_n)^2 + (Olon_n - Dlon_n)^2}. \tag{21}$$

As mentioned in Section 2, the data transmission rate is denoted as λ_{VI}, and then, the number of vehicles and the vehicle set from the original vehicle to the destination vehicle are evaluated by Algorithm 1. In Algorithm 1, the input is the computing task t_n and the destination ECD x_n, and the output is the routing vehicle set, i.e., rs, for computation offloading. The key idea of this algorithm is to obtain the farthest vehicle of v_n that is within the V2V transmission threshold δ when the obtained vehicle is not within the coverage of the x_n-th ECD. Then, this process will be conducted multiple times until the obtained vehicle is within the x_n-th ECD.

Fig. 2 shows an example of acquisition in which there are two ECDs, i.e., d_1 and d_2, and five vehicles, i.e., v_1, v_2, v_3, v_4 and v_5. In this example, assume that the distance between v_1 and v_2 is 5, the distance between v_1 and v_3 is 8,

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Algorithm 1 Route obtaining

Require: The computing task t_n and the destination ECD x_n

Ensure: The routing vehicle set for computation offloading rs

flag = 1
$i = n$
Add v_i to rs
while flag == 1 **do**
  if v_i is not in the coverage of the x_n-th ECD **then**
    $j = i + 1$
    Get the distance g between v_i and v_j by (21)
    $f = 1$
    while $f = 1$ **do**
      Get the distance dis between v_i and v_{j+1}
      if v_{j+1} is not in x_n-th ECD && $dis > g$ && $dis \leq \delta$ **then**
        $j = j$+1
      else $f = 0$
        Add v_j to rs
        $i = j$
      end if
    end while
  else flag = 0
  end if
end while
return rs

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

the distance between v_3 and v_4 is 4, and the distance between v_3 and v_5 is 8. If the coverage distance for V2V communication is 10, when the task from v_1 is transmitted to d_2, the task is chosen to be offloaded to v_3 first. Aiming at reducing task transmission time, the distance between the destination vehicle and v_3 should be minimized. Considering that the distance between v_3 and v_4 is 4 while the distance between v_3 and v_5 is 8, we choose v_4 as the destination vehicle. Thus, the routing vehicle set is $\{v_1, v_3, v_4\}$.

3.2. Computation Offloading Using NSGA-II

An edge computing-enabled computation offloading method with privacy preservation is proposed in this section. The computation offloading problem for IoVs can be defined as a multi-objective optimization problem. NSGA-II has a more accurate and faster global search capability and can solve optimization problems for multiple objectives. Improved mutation can also make the algorithm converge faster and identify the optimal offloading strategy. As the computation offloading problem in this paper is the multi-objective optimization problem, NSGA-II can find the global optimal solution quickly and accurately compared with the traditional genetic algorithm. Thus, NSGA-II is adopted to solve the multi-objective optimization problem presented in (17).

We encode for the ECDs, and fitness functions are given for the optimization problem. The *fast non-dominated sorting* method and the *crowded-comparison* are used in the selection operations. Then, the improved mutation of the genetic algorithm (GA) is adopted. Finally, the overview of the offloading method is elaborated.

3.2.1. Encoding

GA is a population-based method that uses solutions to obtain trade-offs of multi-objective problems. For the computation offloading problems in IoV, a gene represents the offloading strategy for a computing task. A group of genes compose a chromosome, which represents a set of offloading strategies for computing tasks. The value of the offloading strategy is the location of the

ACCEPTED MANUSCRIPT

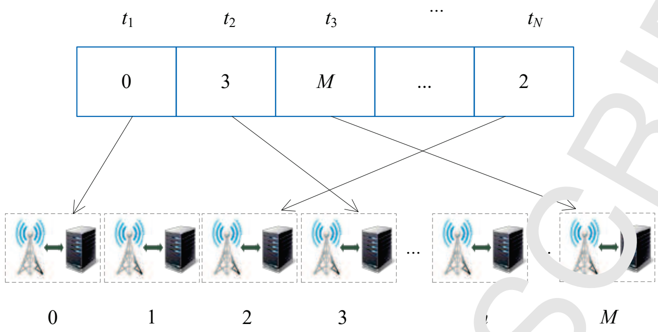


Figure 3: An encoding example of computation offloading for computing tasks.

ECDs and encoded as 0, 1, 2, . . . , M. Fig. 3 shows an example of computation offloading for the computing tasks in T with M ECDs. In this example, the computing tasks are offloaded to ECDs using the result of the offloading strategy, and the codes for t_1, t_2, t_3 . . . and t_n are 0, 3, M, . . . and 2, respectively.

3.2.2. Fitness Functions and Constraints

The fitness function is a criterion for evaluating the possible solutions in GA; each individual represents a solution to problems, and all solutions form the sets, which are called a population. The fitness functions in this paper consist of two parts: (8) and (14), which represent the total time cost and the total energy consumption. The fitness of a solution is the product of trade-offs between two objectives. Both fitness functions must be minimized to measure the performance of this method and the load balancing of resources.

Based on the model we designed in Section II, the objectives are to optimize the total time and reduce energy consumption while also taking into account the capacity of each ECD and privacy preservation for computing tasks. The constraints are given in (18), (19) and (20). NSGA-II provides an effective mechanism to meet the different constraints.

3.2.3. Initialization

In the initialization phase, the parameters should be determined in advance, including the size of the population δ_{pop}, the number of iterations Gen, the

ACCEPTED MANUSCRIPT

crossover probability η_c and the mutation probability η_m. For each offloading strategy, $X^j = [x_1, x_2, \ldots, x_N]$, where X^j represents the j-th chromosome in the population θ. Two populations, C_j and O_j, of size δ_{pop} are randomly generated and mixed together to form a population B_j with a population size of $2\delta_{pop}$.

3.2.4. Selection

The selection operation selects some of the chromosomes for recombination to generate the next population, perform crossover and mutation operations and generate a new population with better fitness.

The population B_j with a population size of $2\delta_{pop}$ generates multiple non-dominated layers $(H_i, i = 1, 2, \ldots)$ using the *fast non-dominated sorting approach*. At the same time, the crowding distance is computed for all individuals in each layer. The selection method in NSGA-II is based on the *crowded-comparison operator*. By calculating the crowding distance of each offloading strategy, the more appropriate individuals can be used to form the elite population for the following operations, as calculated by

$$j_d = j_d^G + j_d^E = |G^{j+1} - G^{j-1}| + |E^{j+1} - E^{j-1}|, \tag{22}$$

where j_d represents the crowding distance of the j-th offloading strategy X^j and j_d^G and j_d^E represents the objective functions. G^{j+1} represents the value of the $j+1$-th offloading strategy to the objective function G.

3.2.5. Crossover and Mutation

The crossover operation aims to combine the two parental chromosomes in the population to obtain better offspring. A crossover point is selected in the chromosome first, two parental genes on both sides of the point are swapped, and then, finally, the crossover operation is completed.

The mutation slightly modifies some of the genes in a chromosome to avoid early convergence. In contrast to the standard mutation operator, we propose an improved mutation in our method. According to the direction of the vehicle, we

ACCEPTED MANUSCRIPT

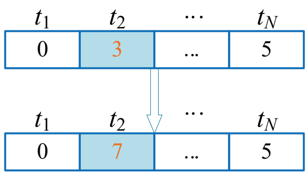


Figure 4: An example of improved mutation operation.

give priority to offloading the computing task to the ECD that is near or in front of its driving direction. Fig. 4 shows the example of an improved mutation in which each gene is changed with equal probability, and t_2 is mutated from 3 to 7. Using this strategy, convergence can be achieved effectively with fewer iterations. After the *Crossover and Mutation operation*, the generated offloading strategy still meets the constraints of privacy preservation.

3.2.6. Method Overview

In this paper, our aim is to achieve the goal of minimizing the execution time and reducing energy consumption. This computation offloading problem is defined as a multi-objective problem, and the improved NSGA-II is used to obtain the optimal computation offloading strategy for IoV. First, the offloading strategies for computing tasks are encoded as the number of vehicles, and fitness functions are given for the computation offloading problems. Then, the *fast non-dominated sorting* approach in NSGA-II is used to generate multiple non-dominated layers for individuals and pretreat the population to better distinguish the merits of individuals. *Crowding distance computation* is used to identify individuals with better fitness. Finally, the improved mutation operation is proposed to accelerate the convergence of the algorithm.

The overview of our method is shown in Algorithm 2. The inputs of algorithm 2 are vehicles, ECDs and computing tasks. The algorithm starts from the first iteration (Line 1). Two populations C_j and O_j of size δ_{pop} are randomly generated and form a population B_j with a population size of $2\delta_{pop}$. The population B_j is divided into multiple non-dominated layers by fast non-dominated sort (Lines 3 and 4). B_j is prepared for the selection operation, and population

ACCEPTED MANUSCRIPT

Algorithm 2 Computation Offloading Method

Require: D, V, T, S, θ

Ensure: Optimal offloading strategy θ^*

    $g = 1$
    while $g \leq Gen$ **do**
        $B_j = C_j + O_j$
        $H = Fast\ non-dominated\ sort\,(B_j)$
        $C_j = \emptyset$
        $j = 1$
        while $num\,(C_j) < \delta_{pop}$ **do**
            Obtain routing vehicles by Algorithm 1
            Calculate crowding distance (H_i) by (22)
            $C_j += H_i$
            $j = j+1$
        end while
        $O_j = Crossover\ and\ mutation\,(C_j)$
        $g = g + 1$
    end while
    return θ^*

C_j is set to empty to store the new generation of the population. In addition, the selection operation follows two rules: first, the higher level of non-dominated layers is prioritized; second, the better crowding distance is prioritized when individuals are in the same non-dominated level. The excellent individuals are selected to fill in a new population of size δ_{pop} by crowding distance computation (Lines 7 to 12). Then, the offspring are generated after the crossover and mutation and put into O_j (Line 13). The offspring population O_j are merged with the parent population C_j and iterated again until the algorithm stops (Lines 2 to 15). Finally, the optimal offloading strategies are output.

ACCEPTED MANUSCRIPT

4. Experimental Evaluation

In this section, a set of comprehensive simulations and experiments are performed to evaluate the performance of the proposed edge computing offloading method, i.e., ECO. Specifically, the simulation setup is introduced first, including the simulation parameter settings and the statements of comparative methods. Then, the influence of different vehicle scales on the time and energy consumption performance of the compared methods and our proposed ECO method is evaluated.

4.1. Simulation Setup

In our simulation, there are some vehicles along a unidirectional road. Six datasets with different scales of vehicles along the road are applied for our experiments, and the number of vehicles is set to 20, 40, 60, 80, 100 or 120. The data transmission rate based on V2V technology, i.e., λ_{V2V}, and the data transmission power, i.e., λ_{V2I}, in our experiment are set to 1 Gb/s and 600 Mb/s according to [4] and [21]. The specified parameter settings in this experiment are illustrated in Table II.

Table 2: Parameter settings

Parameter description	Value
The data transmission rate based on V2V technology λ_{V2V}	1 Gb/s
The total number of ECDs M	20
The power rate of the servers in the ECDs α	300 W
The power rate of the employed resource units β	50 W
The power rate of the employed resource units γ	30 W
The data transmission power λ_{V2I}	600 Mb/s
The processing power of each resource unit p	2000 MHz

To conduct the comparison analysis, we employ some other basic offloading methods with privacy preservation in addition to our ECO method. FFD and

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

BFD are two well-known resource scheduling method while there are still several shortcomings. Therefore, we employ these two methods as comparison methods to evaluate the performance of the proposed method. The comparative methods are briefly expounded as follows.

- Benchmark: A computing task is offloaded to the surrounding ECD that covers the vehicles with the offloading requirements. When the surrounding ECD has no spare space to host extra computing tasks, these tasks are offloaded to the neighbor ECDs. In addition, if the computing tasks have privacy conflicts, they are not offloaded to the same ECD. This process is repeated until all computing tasks are offloaded.

- First Fit Decreasing in Edge computing with Privacy preservation (FFD-EP): The computing tasks are sorted in descending order first according to their requested number of resource units. Then, the sorted computing tasks are offloaded to the surrounding ECDs. If the remaining resources of the ECD are insufficient for hosting any other computing tasks, the new coming computing task is offloaded to another ECD with sufficient resources chosen from the ECD set in order.

- Best Fit Decreasing in Edge computing with Privacy preservation (BFD-EP): The computing tasks and the ECDs are both sorted in descending order according to the computing task resource request and the space of the ECDs first. Then, the sorted computing tasks are offloaded to the sorted ECDs. If the current computing task requires more resources than the current ECD owns, the current computing task is offloaded to the next ECD with sufficient resources. In addition, computing tasks with privacy conflicts are not offloaded to the same ECD.

The methods are implemented under the simulation tools by CloudSim on a PC machine with 2 Intel Core i5-6500U 3.20 GHz processors and 8 GB RAM. The corresponding evaluation results are depicted in detail in the following sections.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4.2. Performance evaluation of ECO

The proposed ECO method aims to achieve trade-offs between optimizing the time consumption and reducing energy consumption. Fifty experiments are conducted in the case of convergence for each vehicle scale, and multiple sets of results are obtained. To identify a set of relatively better solutions, simple additive weighting (SAW) and multiple criteria decision making (MCDM) are used and are measured as follows:

$$V^{'}(c_i) = \frac{1}{2} \cdot \frac{G^{\max} - G(c_i)}{G^{\max} - G^{\min}} + \frac{1}{2} \cdot \frac{E^{\max} - E(c_i)}{E^{\max} - E^{\min}}, \tag{23}$$

where $G(c_i)$ and $E(c_i)$ represent the fitness of the offloading strategy c_i regarding the two objective functions, respectively [2[illegible]23]. G^{max} and G^{min} represent the maximum and minimum fitness for time consumption. If $G^{max} = G^{min}$, let $\frac{G^{\max}-G(c_i)}{G^{\max}-G^{\min}} = 1$. Analogously, E^{max} and E^{min} represent the maximum and minimum fitness for energy consumption; if $E^{max} = E^{min}$, let $\frac{E^{\max}-E(c_i)}{E^{\max}-E^{\min}} = 1$.

4.2.1. Comparison of energy consumption

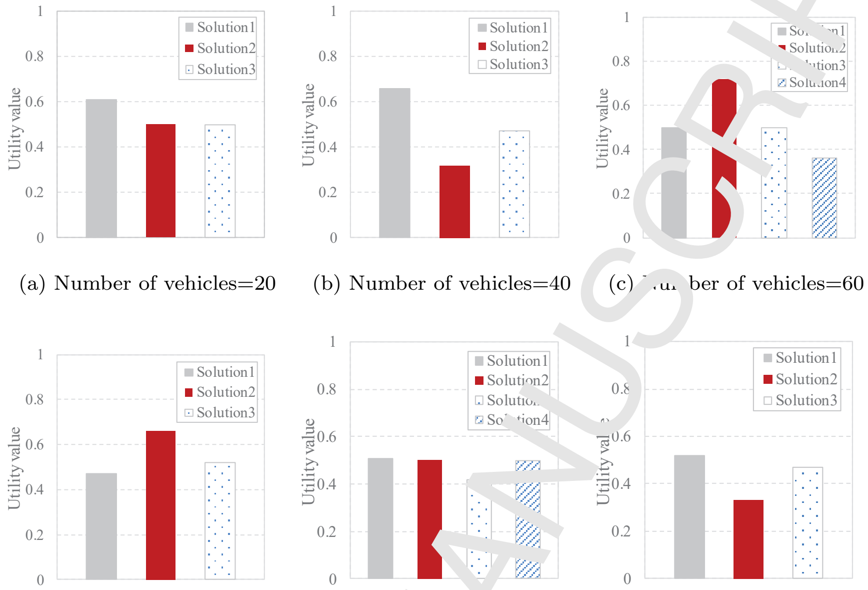
The six sub-figures in Fig. 5 show the comparison of the utility value of the solutions generated by ECO at different vehicle scales. It is intuitive from Fig. 5 that when the vehicle scale is 20, 40, 60, 80, 100 and 120, the number of solutions generated by ECO is 3, 3, [illegible], 3, [illegible] and 3, respectively. For the solutions generated by ECO, we attempt to obtain the most balanced offloading strategy by judging the utility value given in (23). After statistics and analysis, the solution with the maximum utility value is treated as the most balanced strategy. For example, in Fig. 5(a), the final selected strategy is solution 1 because it achieves the highest utility value.

4.3. Comparison analysis

In this subsection, the comparisons of Benchmark, FFD-EP and BFD-EP with the same experimental context are analyzed in detail. The execution time and the energy consumption are the two main metrics for evaluating the performance of the computation methods. Furthermore, the number of employed

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT



(a) Number of vehicles=20 (b) Number of vehicles=40 (c) Number of vehicles=60

(d) Number of vehicles=80 (e) Number of vehicles=100 (f) Number of vehicles=120

Figure 5: Comparison of the utility value of the solutions generated by ECO at different vehicle scales.

ECDs, the resource utilization and the number of computing tasks needed to offload across ECDs are presented to show the real resource usage of all ECDs for hosting the computing tasks. The corresponding results are shown in Figs. 6, 7, 8, 9, 10, 11 and 12.

4.3.1. Comparison of the number of employed ECDs

The number of ECDs employed by the four offloading methods is illustrated in Fig. 6. The total number of ECDs in our experiment is set to 20. As shown in Fig. 6, ECO employs fewer or the same number of ECDs compared to Benchmark, FFD-EP and BFD-EP. In addition, as the number of vehicles increases, the number of ECDs used by ECO increases, and when the number of vehicles reaches 100, all ECDs should be in running mode to respond to the deployment requests of the computing tasks.

ACCEPTED MANUSCRIPT

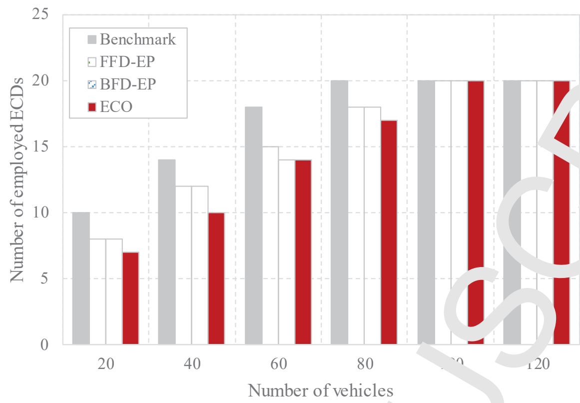


Figure 6: Comparison of the number of ECDs employed at different vehicle scales by Benchmark, FFD-EP, BFD-EP and ECO.

4.3.2. Comparison of resource utilization

After offloading all computing tasks to the ECDs via relevant strategies, the occupation of the resource units is definitely achieved. Fig. 7 shows the comparison of the resource utilization of the ECDs by using Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales. The resource utilization is calculated according to the number of employed ECDs and the employed resource units in each ECD. Fewer employed ECDs with more employed resource units yield a higher resource utilization. It is intuitive from Fig. 7 that our proposed method ECO achieves higher resource utilization than the other three offloading methods. That is, ECO wastes fewer resources than the other methods.

4.3.3. Comparison of the number of offloaded computing tasks across ECDs

In general, the computing task is offloaded to the nearby ECD. However, at small vehicle scales, the location of the vehicle is randomly distributed in different ECD ranges, and if all computing tasks are offloaded to their surrounding ECDs, multiple ECDs will be open, leading to excessive energy consumption. Consequently, in our experiment, the computing task might be offloaded to a neighbor ECD near the surrounding ECD. Offloading computing tasks across

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

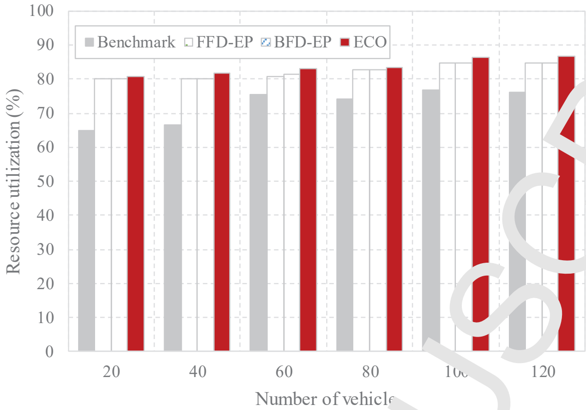


Figure 7: Comparison of the resource utilization of Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

ECDs allows for a computing task to be transferred from the origin vehicle in which the computing task is located in a vehicle in which the coverage of the destination ECD is different from that of the origin ECD. In Fig. 8, we compare the number of computing tasks offloaded across ECDs by the four different computing offloading methods. It is intuitive from Fig. 8 that as the vehicle scale increases, our proposed ECO method transmits more computing tasks across ECDs to achieve better resource utilization.

4.3.4. Comparison of energy consumption

As outlined in Section II, the energy consumption is composed of the baseline energy consumption for all servers in the ECDs, the energy consumption of the employed resource units, and the energy consumption of the unemployed resource units. In Fig. 9, we compare these three aspects of energy consumption at different vehicle scales. As shown in Fig. 9(a), as the vehicle scale increases, all methods increase baseline energy consumption for all servers in the ECDs, but ECO consumes less energy than the other three methods because it employs fewer ECDs. Fig. 9(b) shows that as the number of vehicles increases, the energy cost of employed resource units increases. These four methods achieve the same

ACCEPTED MANUSCRIPT

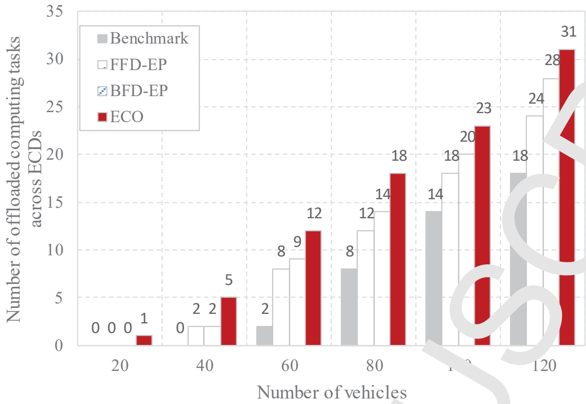


Figure 8: Comparison of the number of offloaded computing tasks across ECDs by Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

energy consumption of the employed resource units at the same vehicle scale because the same number of resource units are employed by Benchmark, FFD-EP, BFD-EP and ECO in the computing tasks. Fig. 9(c) indicates that ECO generates less energy due to unemployed resource units compared to Benchmark, FFD-EP and BFD-EP by using fewer ECDs.

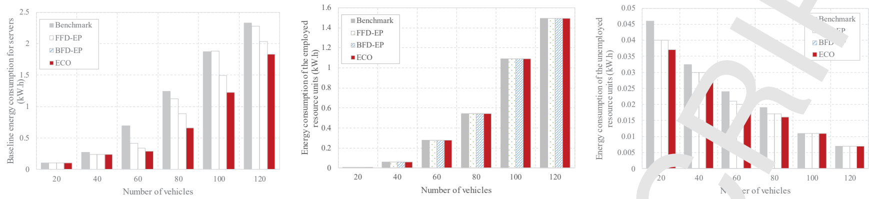
The comparison of energy consumption in Fig. 10 shows that ECO has better performance. For example, when the number of vehicles is 100, ECO achieves a power consumption of less than 2.5 KW.h, whereas Benchmark, FFD-EP and BFD-EP generate more than 2.5 KW.h.

4.3.5. Comparison of time consumption

The offloading time is a fundamental metric of time consumption. Fig. 11 shows the comparison of the offloading times of Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales. It is intuitive from Fig. 11 that our proposed method costs less offloading time than the other methods.

In Fig. 12, we compare the total time consumption of the different offloading methods. It is intuitive that our proposed method ECO costs less time than the compared methods. However, the difference is not obvious when the vehicle

ACCEPTED MANUSCRIPT



(a) energy consumption of servers

(b) energy consumption of the employed resource units

(c) energy consumption of the unemployed resource units

Figure 9: Comparison of different components of energy consumption by Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

scale is small, possibly because ECO needs to transmit more offloaded computing tasks across the ECDs than the other three methods, which consumes more transmission time. As the scale of the vehicles increases, the influence of transmission time becomes small.

5. Related Work

Over the past few years, MEC, which has a faster data processing rate and more stable transmission, has undergone a tremendous revolution as a new computing paradigm in the IoV environment [24][25][26]. Moreover, offloading computationally intensive workloads to ECDs reduces energy consumption and delays, enhancing the quality of computation.

As a multidisciplinary ecosystem, IoV is connected to scenarios that demand real-time data processing and feedback. However, traditional cloud platforms are not suitable for scenarios requiring real-time processing, low latencies, and a high-quality computing experience. Due to the delays and unstable connections associated with remote clouds, MEC is more suitable in the IoV environment. Compared with traditional cloud computing, MEC provides computing resources and extra storage closer to vehicles and end users [27][28][29].

Given the rapid increase in IoV applications, it is of great urgency to design a MEC architecture that can adequately process large quantities of data for

ACCEPTED MANUSCRIPT

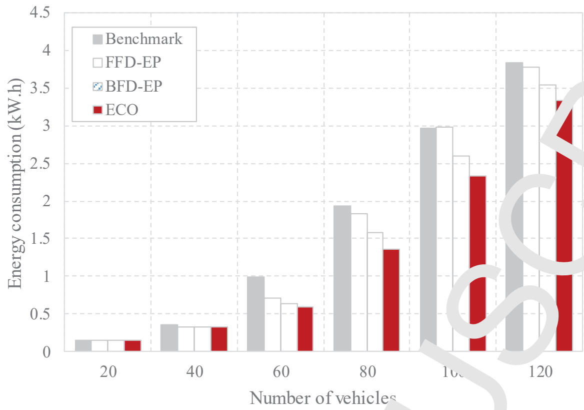


Figure 10: Comparison of the energy consumption by Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

vehicles. In [30], Hu et al. proposed a multi-access edge computing framework as well as the corresponding communication protocol. To process and distribute the contents efficiently, the proposal integrated various technologies such as a licensed Sub-6 GHz band and millimeter wave communications. Similarly, by integrating different types of technologies, Liu et al. proposed an SDN (software-defined network)-enabled network architecture in [31]. The MEC algorithm has the on-premises features, which decreases execution time and enhances the quality of the experience, and could be utilized to perfect the architecture while ensuring satisfying scalability and responsiveness. However, although ECDs can perform the task of processing data with low latency in MEC, the computing resource deployed in MEC are restricted. Hence, allocating and coordinating resources between the edge and cloud servers is a necessity. In [32], Sasaki et al. proposed an infrastructure-based vehicle control system. With the system, resources and computations are allocated dynamically based on the data collected by sensors. In [33], Kumar et al. proposed an architecture in which complex computations are performed by devices located at the edge of the computation in light of the high mobility of vehicles.

Generally, computation offloading refers to offloading complex workloads to

ACCEPTED MANUSCRIPT

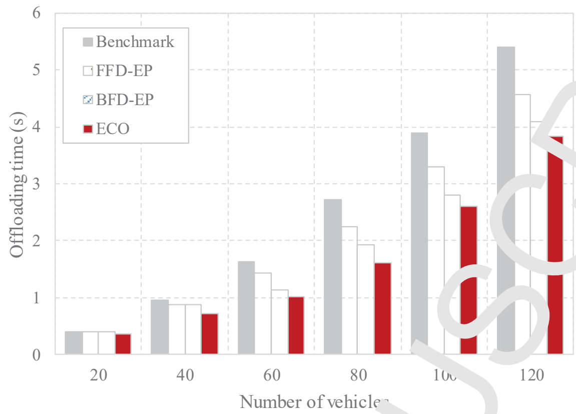


Figure 11: Comparison of the offloading time consumption of Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

servers with the required resources. As a practical part of MEC, offloading was originally intended to offload tasks on demand. By offloading heavy tasks to ECDs, lower latencies and lower energy consumption could be achieved, improving the quality of the computing experience [34][35][36].

In [37], Mach et al. provided an overview of several principles in terms of offloading, including classification, influencing factors and management in practice. Based on these principles, they sorted efforts to address the challenges of whether to offload and how to allocate computing resource. In [38], Mao et al. proposed a low-complexity sub-optimal algorithm to optimize task offloading scheduling and allocate transmit power legitimately. As advantages of alternating minimization, the weighted sum of the delay and energy consumption are able to reach the minimum. In addition, convex optimization techniques are utilized for the transmit power allocation under a given offloading scheduling decision. In addition, in [39], Sardellitti et al. proposed an iterative algorithm based on convex optimization. They formulated the offloading problem as the reduction in energy consumption and latency, and thus, the result of the optimization problem is nonconvex in the multi-users case and can be obtained by

ACCEPTED MANUSCRIPT

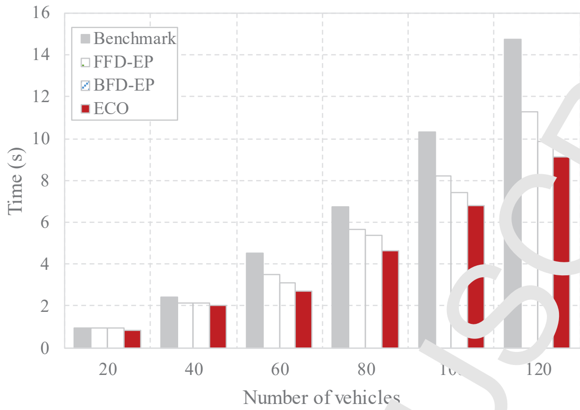


Figure 12: Comparison of the time consumption of Benchmark, FFD-EP, BFD-EP and ECO at different vehicle scales.

the algorithm. In [40], for a wireless powered multiuser MEC system, Wang et al. proposed a unified MEC-WPT design in which computing tasks could be executed locally by broadcasting wireless power to multiple users. Furthermore, a framework was developed to optimize energy consumption and execution time. In [41], Chen et al. proposed a distributed computation offloading algorithm in the multi-user computation offloading problem with the aim of achieving a Nash equilibrium and quantifying efficiency from the aspect of performance metrics.

As the IoV achieves the persuasive milestones over the last couple of decades, it is a trend to equip the vehicles with the ability of intelligent driving, giving rise to the communication between vehicles [42][43]. The communication is useful in many ways, especially in driving security and privacy protection [44][45][46]. In addition to privacy leaks, virtual vehicle hijacking is more serious. Although intellectualizing the operations of vehicles, the use of sensors makes it easy to invade the vehicles' electronic systems. If the systems are invaded, the schemers could inject incorrect orders and transmit false information to the destination vehicle [47][48]. Consequently, it is of urgency to prepare in advance for virtual vehicle hijacking, such as establishing a trust judgement method and designing a dual authentication scheme on different scenarios [48][49].

ACCEPTED MANUSCRIPT

Few studies have examined multi-objective optimization for offloading computing tasks from vehicles across ECDs in an IoV environment. Achieving the goals of energy conservation and transmission delay reduction while satisfying privacy protection constraints remain challenging. Therefore, an offloading method is proposed to address the above challenge in this paper.

6. Conclusion and Future Work

In recent years, IoV has emerged as a powerful technology for providing real-time traffic information to drivers and transportation control systems. With the rapid development of IoV technology, computing tasks become so complex that it is necessary to offload the tasks to the remote infrastructure. The MEC paradigm is one of the most effective paradigms in terms of processing IoV computing tasks, in which the computing tasks of the vehicles are offloaded to ECDs in close proximity to the vehicles. To realize multi-objective optimization to reduce the execution time of the computing tasks and the energy consumption of the ECDs while satisfying the privacy conflicts of the computing tasks, an edge computing-enabled computation offloading method named ECO is proposed in this paper. First, to acquire the routing vehicles from the origin vehicle in which the computing task is located to the destination vehicle, V2V communication-based routing for a vehicle is developed. Then, NSGA-II is utilized to achieve the multi-objective optimization. Subsequent experimental evaluations verify the efficiency and effectiveness of ECO.

In future work, we will attempt to adapt and extend our proposed method to a real-world scenario of IoV services, and we will specify the different time requirements of the computing tasks to attempt to identify an offloading strategy to achieve energy savings of ECDs with definite time constraints as well as privacy constraints.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

7. Acknowledgment

This research is supported by the National Natural Science Foundation of China under grant no. 61702277, no. 61472024, no. 61432008 and no. 61872219. This work is also supported by The Startup Foundation for Introducing Talent of NUIST, the open project from the State Key Laboratory for Novel Software Technology, Nanjing University, under grant no. KFKT2017B04, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

References

[1] J. Contreras, S. Zeadally, J. A. Guerrero-Ibanez, Internet of vehicles: Architecture, protocols, and security, IEEE Internet of Things Journal PP (99) (2017) 1–1.

[2] N. Kumar, J. J. Rodrigues, N. Chilamkurti, Bayesian coalition game as-a-service for content distribution in internet of vehicles, IEEE Internet of Things Journal 1 (6) (2014) 544–555.

[3] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, M. J. Deen, A distributed hosvd method with its incremental computation for big data in cyber-physical-social systems, IEEE Transactions on Computational Social Systems 5 (2) (2018) 481–492.

[4] L. Kong, M. K. Khan, F. Wu, G. Chen, P. Zeng, Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges., IEEE Communications Magazine 55 (1) (2017) 62–68.

[5] 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.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

[6] R. Deng, H. Liang, Whether to charge or discharge an electric vehicle? an optimal approach in polynomial time, in: Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th, IEEE, 2017, pp. 1–5.

[7] K. Zhang, Y. Mao, S. Leng, Y. He, Y. Zhang, Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading, IEEE Vehicular Technology Magazine 12 (2) (2017) 36–44.

[8] J. Kang, R. Yu, X. Huang, Y. Zhang, Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles, IEEE Transactions on Intelligent Transportation Systems PP (99) (2017) 1–11.

[9] J. Ren, J. Hu, R. Deng, D. Zhang, Y. Zhang, X. S. Shen, Joint load scheduling and voltage regulation in the distribution system with renewable generators, IEEE Transactions on Industrial Informatics 14 (4) (2018) 1564–1574.

[10] F. Qu, Z. Wu, F. Y. Wang, W. Cho, A security and privacy review of vanets, IEEE Transactions on Intelligent Transportation Systems 16 (6) (2015) 2985–2996.

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646.

[12] D. He, S. Zeadally, B. Xu, X. Huang, An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks, IEEE Transactions on Information Forensics and Security 10 (12) (2015) 2681–2691.

[13] S. Wan, Y. Zhang, J. Chen, On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks, IEEE Sensors Journal 16 (20) (2016) 7433–7440.

[14] Y. Qian, M. Chen, J. Chen, M. S. Hossain, A. Alamri, Secure enforcement in cognitive internet of vehicles, IEEE Internet of Things Journal PP (99) (2018) 1–1.

ACCEPTED MANUSCRIPT

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

[16] Y. Mao, J. Zhang, K. B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE Journal on Selected Areas in Communications 34 (12) (2016) 3590–3605.

[17] X. Xu, S. Fu, L. Qi, X. Zhang, Q. Liu, Q. He, S. Li, An iot-oriented data placement method with privacy preservation in cloud environment, Journal of Network and Computer Applications 124 (2018) 148–157.

[18] J. Lee, H. Yu, Y. Sung, Beam tracking for interference alignment in time-varying mimo interference channels: A conjugate-gradient-based approach, IEEE Transactions on Vehicular Technology 63 (2) (2014) 958–964.

[19] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, K.-K. R. Choo, Multi-dimensional data indexing and range query processing via voronoi diagram for internet of things, Future Generation Computer Systems 91 (2019) 382–391.

[20] Z. Gao, D. Wang, S. Wan, H. Zhang, Y. Wang, Cognitive-inspired class-statistic matching with triple-constrain for camera free 3d object retrieval, Future Generation Computer Systems.

[21] G. Premsankar, B. Ghaddar, M. Di Francesco, R. Verago, Efficient placement of edge computing devices for vehicular applications in smart cities, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018.

[22] E. K. Zavadskas, Z. Turskis, S. Kildien, State of art surveys of overviews on mcdm/madm methods, Technological & Economic Development of Economy 20 (1) (2014) 165–179.

ACCEPTED MANUSCRIPT

[23] I. Kaliszewski, D. Podkopaev, Simple additive weightinga metamodel for multiple criteria decision analysis methods, Expert Systems with Applications 54 (C) (2016) 155–161.

[24] K. M. Alam, M. Saini, A. El Saddik, Toward social internet of vehicles: Concept, architecture, and applications, IEEE access 3 (2015) 343–357.

[25] S. Wang, C. Fan, C.-H. Hsu, Q. Sun, F. Yang, A vertical handoff method via self-selection decision tree for internet of vehicles, IEEE Systems Journal 10 (3) (2016) 1183–1192.

[26] X. Wang, L. T. Yang, H. Liu, M. J. Deen, A big data-as-a-service framework: State-of-the-art and perspectives, IEEE Transactions on Big Data PP (99) (2017) 1–1.

[27] C. You, K. Huang, H. Chae, B.-H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Transactions on Wireless Communications 16 (3) (2017) 1397–1411.

[28] C. F. Cheang, Y. Wang, Z. Cai, G. Xu, Multi-vms intrusion detection for cloud security using dempster-shafer theory PP (99) (2018) 1–1.

[29] R. Deng, R. Lu, C. Lai, T. H. Luan, Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing, in: IEEE International Conference on Communications, 2015, pp. 3909–3914.

[30] Q. Hu, C. Wu, X. Zhao, X. Chen, Y. Ji, T. Yoshinaga, Vehicular multi-access edge computing with licensed sub-6 ghz, ieee 802.11 p and mmwave, IEEE Access 6 (2018) 1995–2004.

[31] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, M. Qiu, A scalable and quick-response software defined vehicular network assisted by mobile edge computing, IEEE Communications Magazine 55 (7) (2017) 94–100.

[32] K. Sasaki, N. Suzuki, S. Makido, A. Nakao, Vehicle control system coordinated between cloud and mobile edge computing, in: Society of Instrument

and Control Engineers of Japan (SICE), 2016 55th Annual Conference of the, IEEE, 2016, pp. 1122–1127.

[33] 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.

[34] B. P. Rimal, D. P. Van, M. Maier, Mobile-edge computing vs. centralized cloud computing in fiber-wireless access networks, in: Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on, IEEE, 2016, pp. 991–996.

[35] V. Gopinath, R. Bhuvaneswaran, Design of ecc based secured cloud storage mechanism for transaction rich applications PP (99) (2018) 1–1.

[36] R. Deng, H. Liang, J. Yong, B. Chai, T. Yang, Distributed rate control, routing, and energy management in dynamic rechargeable sensor networks, Peer-to-Peer Networking and Applications 10 (3) (2016) 1–15.

[37] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, arXiv preprint arXiv:1702.05309 PP (99) (2017) 1–11.

[38] Y. Mao, J. Zhang, K. B. Letaief, Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems, in: Wireless Communications and Networking Conference (WCNC), 2017 IEEE, IEEE, 2017, pp. 1–6.

[39] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Transactions on Signal and Information Processing over Networks 1 (2) (2015) 89–103.

[40] F. Wang, J. Xu, X. Wang, S. Cui, Joint offloading and computing optimization in wireless powered mobile-edge computing systems, IEEE Transactions on Wireless Communications 17 (3) (2018) 1784–1797.

ACCEPTED MANUSCRIPT

[41] X. Chen, L. Jiao, W. Li, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Transactions on Networking (5) (2016) 2795–2808.

[42] H. Liu, Y. Zhang, T. Yang, Blockchain-enabled security in electric vehicles cloud and edge computing, IEEE Network 32 (3) (2018) 78–83.

[43] X. Wang, L. T. Yang, X. Xie, J. Jin, M. J. Deen, A cloud-edge computing framework for cyber-physical-social services, IEEE Communications Magazine 55 (11) (2017) 80–85.

[44] M. K. Afzal, B. S. Kim, S. W. Kim, Efficient and reliable mpeg-4 multicast mac protocol for wireless networks, Vehicular Technology IEEE Transactions on 64 (3) (2015) 1026–1035.

[45] L. T. Yang, X. Wang, X. Chen, L. Wang, R. Ranjan, X. Chen, M. J. Deen, A multi-order distributed hosvd with its incremental computing for big services in cyber-physical-social systems, IEEE Transactions on Big Data PP (99) (2018) 1–1.

[46] S. Zhang, K.-K. R. Choo, Q. Liu, G. Wang, Enhancing privacy through uniform grid and caching in location-based services, Future Generation Computer Systems 86 (2018) 881–892.

[47] A. Dua, N. Kumar, A. K. Das, W. Susilo, Secure message communication protocol among vehicles in smart city, IEEE Transactions on Vehicular Technology PP (99) (2018) 1–1.

[48] Y. Liu, Y. Wang, G. Chang, Efficient privacy-preserving dual authentication and key agreement scheme for secure v2v communications in an iov paradigm, IEEE Transactions on Intelligent Transportation Systems PP (99) (2017) 1–10.

[49] X. Wang, C. Xiong, Q. Pei, Y. Qu, Expression preserved face privacy protection based on multi-mode discriminant analysis, CMC: computers, materials & continua 57 (1) (2018) 107–121.

ACCEPTED MANUSCRIPT

1. Xiaolong Xu received his Ph.D. degree from the Nanjing University, China, in 2016. He worked as a research scholar at Michigan State University, USA, from Apr. 2017 to May 2018. He is currently an assistant professor with the school of computer and software, Nanjing University of Information Science and Technology. He has published more than 40 peer review papers in international journals and conferences. His research interests include mobile computing, edge computing, IoT, cloud computing and big data.

2. Yuan Xue is currently working towards his B.S. degree in Computer Science and Technology at School of Computer and Software in Nanjing University of Information Science and Technology. His research interests include Big Data and Mobile Cloud Computing.

3. Lianyong Qi received his PhD degree in Department of Computer Science and Technology from Nanjing University, China, in 2011. Now, he is an associate professor of the School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, China. He has already published more than 30 papers including JSAC, TCC, TBD, FGCS, JCSS, CCPE. His research interests include recommender systems and services computing.

4. Yuan Yuan is currently a Postdoctoral Research Fellow with the Department of Computer Science and Engineering, Michigan State University, USA. He received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2015. From 2014 to 2015, he was a visiting Ph.D. student with the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, U.K. He worked as a Research Fellow at the School of Computer Science and Engineering, Nanyang Technological University, Singapore, from 2015 to 2016. His research interests include evolutionary computation, machine learning, and search-based software engineering.

5. Xuyun Zhang is a lecturer in the Department of Electrical & Computer Engineering at the University of Auckland, New Zealand. Prior to his current appointment, he worked as a postdoctoral fellow in the Machine Learning Research Group of NICTA (currently Data61, CSIRO) in Australia. He received his PhD degree from University of Technology, Sydney (UTS, Australia) in 2014, as well as his ME and BS degrees in Computer Science from Nanjing University (China) in 2011 and 2008, respectively. His primary research interests include IoT and smart cities, big data, cloud computing, scalable machine learning & data mining, data privacy & security, and Web service technology.

6. Triq Umer received the Ph.D. degree in communication systems from the School of Computing and Communication, Lancaster University, U.K., in 2012. He served for the IT education sector in Pakistan for over 13 years. Since 2007, he has been an Assistant Professor with the CS Department, COMSATS Institute of Information Technology, Wah Cantt. His research interests include cognitive radio ad hoc networks, Internet of Things, wireless sensor networks, telecommunication network design, vehicular adhoc networks, and Internet of Vehicles. He is currently serving as an Editorial Board Member of Future Generation Computer System and as an Associate Editor of the IEEE Access. He served as a TPC for the international conference FIT 15, 16, IEEE PGnet 2011, 2012, and IEEE FMEC 2016. He is also serving as a Guest Editor of Future

ACCEPTED MANUSCRIPT

Generation Computer Systems and IEEE Access. He is the Active Member of the Pakistan Computer Society and the Internet Society Pakistan.

7. Shaohua Wan received the Ph.D. degree from School of Computer, Wuhan University in 2010. From 2015, he worked as a postdoc at State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology. From 2016 to 2017, he worked as a visiting scholar at Department of Electrical and Computer Engineering at Technical University of Munich, Germany. At present, he is an associate professor at School of Information and Safety Engineering, Zhongnan University of Economics and Law. His research interests include massive data computing for Internet of Things and edge computing.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1. Xiaolong Xu

2. Yuan Xue

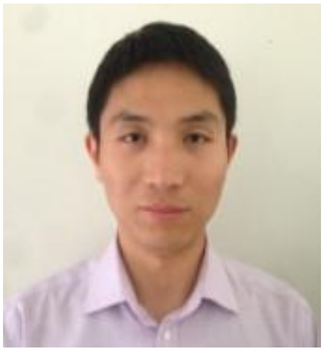
3. Lianyong Qi

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4. Yuan Yuan

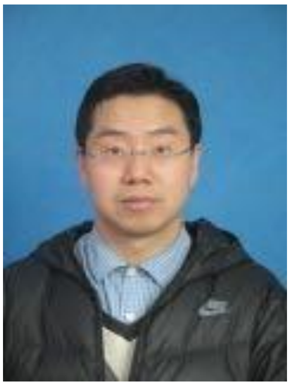
5. Xuyun Zhang



6. Triq Umer

ACCEPTED MANUSCRIPT

7. Shaohua Wan



ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

HIGHLIGHTS

1. Analyze privacy conflicts of the computing tasks offloaded to the edge computing devices.
2. Design a vehicle-to-vehicle communication-based route-obtaining algorithm.
3. Adopt NSGA-II to realize multi-objective optimization while guarding against privacy conflicts.

ACCEPTED MANUSCRIPT