# A Block-structured Optimization Approach for Data Sensing and Computing in Vehicle-assisted Edge Computing Networks

Luan N. T. Huynh, Md. Delowar Hossain, Quoc-Viet Pham, *Senior Member, IEEE,*
Yan Kyaw Tun, *Member, IEEE,* and Eui-Nam Huh, *Member, IEEE*

***Abstract*—With the rapid development of IoT applications and multi-access edge computing (MEC) technology, massive amounts of sensing data can be collected and transmitted to MEC servers for rapid processing. On the other hand, as the number of IoT devices grows, the MEC server cannot perform tremendous computing tasks because of its limited computation capacity. This paper introduces a vehicle-assisted MEC framework that leverages vehicles to provide computational services for IoT devices and overcome this challenge. The problem of latency minimization was formulated by optimizing the sensing data rate, offloading decisions, and resource allocation while considering energy consumption constraints. Nevertheless, achieving the global optimal solution in polynomial time is challenging because the formulated problem is mixed-integer nonlinear and non-convex. This paper provides an efficient algorithm that adopts the block coordinate descent technique to decompose the original problem into four subproblems. These subproblems can be solved using Lagrangian relaxation and the block successive upper-bound minimization method. The superiority of the proposed approach in reducing latency compared to baseline schemes is evident from the simulation results.**

***Index Terms*—Computation offloading, data sensing, edge computing, resource allocation, vehicular network.**

## I. Introduction

INTERNET of things (IoT) devices are composed of a large number of sensors. Their effectiveness is increasing continuously, resulting in a diversity of intelligent IoT applications (e.g., smart home, autonomous driving, augmented reality, and healthcare) [1], [2]. This rapid growth has posed numerous challenges for traditional cloud computing systems regarding latency, bandwidth limitations, and energy consumption. Nevertheless, IoT devices often have limited computing capacity and batteries, making running specific applications that need extensive processing difficult. Traditional cloud computing models are being augmented with edge computing paradigms, which bring computational resources closer to the network edge to meet these requirements. In this context, MEC has emerged as a promising solution by deploying computational and storage capabilities at the network periphery, enabling the execution of time-sensitive applications, and reducing the burden on the centralized cloud. MEC provides computing services, enabling the offloading of tasks from IoT devices to the MEC servers stationed at the network periphery, greatly relieving the energy limitations of the devices [3], [4].

IoT devices can be used to sense the physical world by taking advantage of the sensors available [4]. With the rapid development of various IoT applications, massive amounts of data are gathered by sensing and transferred to servers for further processing [5]. Previous research has explored the integration of data sensing and offloading within MEC systems. These studies aimed to optimize various aspects, such as data throughput, average sensing rates, energy consumption, and data utility in the context of multiuser MEC and crowd sensing scenarios [6], [7]. Some research has focused on reducing energy consumption by fine-tuning parameters like compression ratios, sensing data sizes, and wireless power allocation [8].

On the other hand, the computational capability of the MEC server is restricted compared to that of a centralized cloud. Nevertheless, many devices uploading data to the server can cause overload and increase latency. Furthermore, the extensive deployment of MEC servers could result in resource inefficiency during off-peak times in scenarios where the number of IoT devices grows substantially in specific places and times, such as outdoor events. One of the potential solutions for the limited computational capability issue is vehicle-assisted MEC to extend the edge computing capabilities of conventional MEC. Vehicle-assisted MEC networks introduce a unique dimension to the optimization task offloading problem, leveraging the mobility of vehicles to serve as MEC servers or relays. Vehicle-assisted MEC is a paradigm where the underutilized resources of nearby vehicles are harnessed to assist MEC servers in the execution of offloading tasks [9]. These networks can enhance the performance of MEC systems, particularly in highly dynamic and dense urban environments, using the computational and communication resources available in vehicles [10]. This solution requires no additional infrastructure design or significant additional costs.

Nevertheless, prior studies have not specifically addressed the integration of IoT sensing within the domain of vehicular networks. The present study introduces an innovative approach that harnesses the computing power of vehicles to enhance

This research was partly supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2023-00258649) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program (IITP-2023-RS-2023-00266615) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). (Corresponding author: Eui-Nam Huh.)

Luan N. T. Huynh is with the Institute of Engineering Technology, Thu Dau Mot University and also with the Department of Computer Science and Engineering, Kyung Hee University, e-mail: luanhnt@tdmu.edu.vn.

Md. Delowar Hossain and Eui-Nam Huh are with the Department of Computer Science and Engineering, Kyung Hee University, e-mail: {delowar, johnhuh}@khu.ac.kr.

Quoc-Viet Pham is with the School of Computer Science and Statistics, Trinity College Dublin, D02 PN40, Ireland, e-mail: viet.pham@tcd.ie.

Yan Kyaw Tun is with the Department of Electronic Systems, Aalborg University, A. C. Meyers Vænge 15, 2450 København, e-mail: ykt@es.aau.dk.

MEC capabilities, focusing primarily on IoT devices equipped with sensors. In contrast to previous studies, the present study focuses on minimizing latency by optimizing the data sensing rate, offloading decision, and resource allocation in vehicle-assisted MEC. In summary, the contributions of this work can be summarized as follows:

- First, vehicle-assisted edge computing networks were considered, and a data sensing and offloading scheme was proposed for this system. The latency of IoT devices was decreased by mathematically formulating a joint problem of data sensing rate, offloading fraction, offloading decision, and resource allocation (i.e., bandwidth allocation) for latency minimization subject to various resource and energy consumption constraints.
- Second, as the problem formulated falls into mixed-integer nonlinear programming (MINLP), which is inherently NP-hard, achieving an optimal solution within a polynomial time frame presents a substantial challenge. This issue was addressed by developing an efficient algorithm that decomposes the formulated problem into multiple subproblems and solves them through the Lagrangian relaxation method and the block successive upper-bound minimization (BSUM) method. The proposed method can converge to an optimal solution.
- Finally, the simulation outcomes affirm the effectiveness of the proposed algorithms in converging toward the final solution. In addition, the results obtained from the simulations suggest that the introduced algorithms outperformed the baseline scheme, particularly in terms of latency reduction.

The subsequent sections of this paper were structured as outlined below. Section II provides an overview of existing research on computation offloading and optimization within MEC networks. Section III outlines the network model and the formulation of the problem. The proposed algorithms and optimization strategies are expounded upon in Section IV. The simulation results and their analysis are detailed in Section V. Finally, Section VI provides the conclusions and discusses future research directions.

## II. Related Works

The primary focus of the studies lies in MEC systems, particularly optimizing computational offloading and resource allocation. The objective of these studies is to reduce both completion latency and energy consumption. Ndikumana *et al.* [2] examined integrating computing, communication, control, and caching (4C) in big data MEC systems to mitigate user latency and optimize the backhaul bandwidth utilization. Aghapour *et al.* [11] introduced an offloading strategy based on deep reinforcement learning (DRL) to optimize energy consumption and processing time concurrently. An intelligent offloading mechanism was introduced based on a deep deterministic policy gradient (DDPG) scheme for a multi-user scenario [12]. The system explicitly addresses an optimization challenge to minimize the total energy consumption for multiple tasks within a multi-user context. This optimization accounts for factors, such as computing resource allocation, power split ratio, uplink channel bandwidth, and task offloading ratio to achieve optimal energy consumption and delay. Wang *et al.* [13] focused on minimizing the total cost by optimizing the computational offloading decisions and resource allocation regarding the time delay and the charging aspects. This task was accomplished through optimized decision-making for offloading and content-caching strategies. A previous study [14] focused on the computation offloading and allocation of resource challenges within the NOMA-MEC system. They proposed the DRL technique to reduce the overall computational overhead compared to other baseline methods. Wu *et al.* [15] proposed an efficient algorithm for optimizing workload, offloading, and downloading duration.

Some studies combined data sensing and offloading in MEC systems. Liang *et al.* [6] studied coordinating offloading, data sensing, and MEC server computation within a multi-user system. Their objective was to maximize throughput. The concept of data sensing has been discussed elsewhere [7], where the authors optimized the long-term average sensing rate of the wireless device, considering multiple factors. These factors include maintaining data queue stability for both the MEC server and the wireless device, adhering to the average power restriction of the MEC server, and ensuring the quality of service (QoS) threshold for the primary link. Another study [8] introduced the concept of wirelessly powered crowd sensing (WPCS), which combined mobile crowd sensing (MCS) with wireless power transfer to tackle issues related to battery life and user incentives. The study focused on optimizing a multi-user WPCS system to minimize energy consumption and maximize data utility. This optimization involved fine-tuning various parameters, including compression ratios, sizes of sensing data, and wireless power allocation. Zhou *et al.* [5] addressed the challenges in MCS by optimizing the sensing and transmission rates to reduce the energy used within a single-user MCS system. The proposed approach decomposes the problem into subproblems and employs efficient algorithms for optimal solution finding.

The last few years have witnessed several research works dedicated to resolving vehicle-assisted MEC problems [9], [10], [16]–[19]. Parked vehicle edge computing (PVEC) was introduced to address task allocation challenges using the resources from parked vehicles [9]. The container-based virtualization was integrated with PVEC to achieve flexible and fine-grained resource utilization, ensuring rapid response, scalability, and efficiency in task execution on parked vehicles. Moreover, their approach optimized social welfare for users and parked vehicles simultaneously. Huang *et al.* [10] proposed the concept of parked vehicle edge computing (PVEC) to allocate workloads among parked vehicles and minimize user costs.

On the other hand, parking edge computing using parked vehicles to aid edge servers in handling offloaded tasks was proposed [16]. A task scheduling algorithm that concurrently decides resource allocation and edge server selection was designed to optimize task offloading performance. Furthermore, they developed a predictive model based on the random forest approach to improve the accuracy of output result transmission. A collaborative approach that combines MEC and cloud

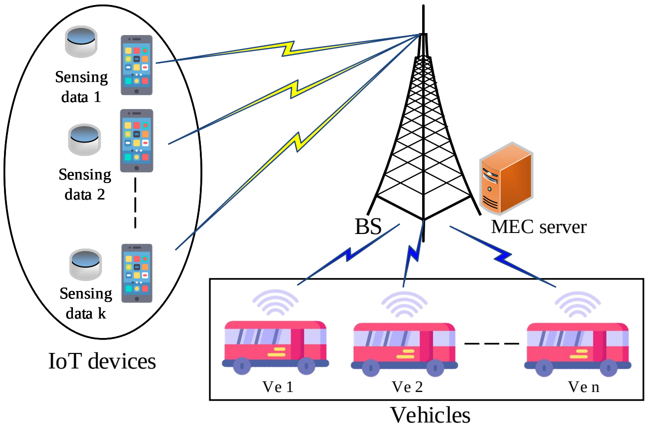


Fig. 1: Illustration of system model.

computing for computation offloading in vehicular networks was introduced [18], addressing a cloud-MEC collaborative computation offloading problem by optimizing decision and resource allocation. Another study [19] considered scenarios where users can connect directly to the MEC server or vehicles for task offloading. The primary aim of this study was to optimize the overall utility of the system. These papers do not specifically consider IoT devices equipped with sensors in the context of vehicular networks. This paper introduced a novel approach that leverages the computational capabilities of vehicles to enhance MEC, focusing mainly on IoT devices with sensors. This study examined the minimization of latency in vehicle-assisted MEC by optimizing the data sensing rate, offloading decision, and allocation of resources.

## III. Network Model and Problem Formulation

### A. Network Model

As shown in Fig. 1, this study examined the MEC system consisting of one base station (BS) installed with an MEC server, a set $\mathcal{K} = \{1, 2, \ldots, k, \ldots, K\}$ as IoT devices, and a set $\mathcal{N} = \{1, 2, \ldots, n, \ldots, N\}$ as vehicles. Each IoT device collects a large amount of sensing data, partially executed locally and offloaded to the BS for remote processing. Owing to the constrained computational capability of the MEC server, the computation task can be transferred from the BS to the vehicles for task execution. This study focused on sequential sensing [8], local computing and offloading, and remote computing. The entire procedure was separated into four stages. This study neglects to download the computational result back to the IoT devices (i.e., downloading latency and energy) as the computation result data size is significantly smaller than that of the input computational data.

### B. Task Data Sensing Model

Denoting the sensing data size $D_k$, the sensing time $T_k^{\text{sen}}$ of the IoT device $k$ can be expressed as follows:

$$T_k^{\text{sen}} = \frac{D_k}{s_k}. \quad (1)$$

where the $s_k$ data sensing rate of IoT device $k$ [6].

According to the work in [5], the energy consumption on data sensing for IoT device $k$ can be obtained as

$$E_k^{\text{sen}} = \kappa (s_k)^2 D_k (W_k)^2. \quad (2)$$

where $\kappa$ is a constant determined by the CPU chip architecture, established at $\kappa = 5 \times 10^{-27}$. $W_k$ is the number of CPU cycles required for sensing a single bit of data.

### C. Local Computation Model

Let $\alpha_k \in [0, D_k]$ denote the fraction of task sensing data of IoT device $k$ that is offloaded to the BS. Thus, $(D_k - \alpha_k)$ is the fraction of task sensing data processed locally. The local computing capability of user $k$ is denoted by $f_k^{\text{loc}}$. The local processing latency of user $n$ is represented as

$$T_k^{\text{loc}} = \frac{(D_k - \alpha_k) C_k}{f_k^{\text{loc}}}. \quad (3)$$

where $C_k$ signifies the number of CPU cycles required for processing a single bit of data. The energy consumption linked to the local processing of the IoT device $k$ can be calculated as follows:

$$E_k^{\text{loc}} = \kappa (D_k - \alpha_k) C_k (f_k^{\text{loc}})^2. \quad (4)$$

### D. Remote Computation Model

Each IoT device can offload a fraction of task sensing data to the BS. Therefore, it is essential to consider wireless channel access in the uploading and downlink processes. Thus, orthogonal frequency-division multiple access (OFDMA) was used for the proposed system model. The uplink transmission rate from IoT device $k$ to BS is defined as

$$R_k^{0,\text{ul}} = \beta_k B^{\text{ul}} \log_2 \left( 1 + \frac{h_k p_k}{\sigma^2} \right), \quad (5)$$

where $h_k$ represents the channel gain between IoT device $k$ and BS; $p_k$ is the transmission power of IoT device $k$ during offloading; $\beta_k$ is the proportion of bandwidth allocated to offloading of IoT devices $k$; $B^{\text{ul}}$ is the total bandwidth for uplink; $\sigma^2$ denotes the additive Gaussian white noise (AGWN) power.

The uplink transmission latency for uploading the task sensing data of the IoT device $k$ to BS can be defined as

$$T_k^{0,\text{ul}} = \frac{\alpha_k}{R_k^{0,\text{ul}}}. \quad (6)$$

Therefore, the energy consumption of IoT device $k$ for offloading the data to BS is given as

$$E_k^{0,\text{ul}} = p_k T_k^{0,\text{ul}}. \quad (7)$$

The binary variable $x_k \in \{0, 1\}$ was established as a decision parameter, where $x_k = 1$ if the offloaded sensing task of IoT device $k$ is processed at the BS and $x_k = 0$, otherwise.

The maximum CPU frequency of the MEC server is given by $f^{max}$. The CPU frequency allocated to handle the offloaded task of the IoT device $k$ at the MEC server is represented as $f_k$, i.e., $f_k$, the following constraint when allocating the computing capacity of the MEC server must be satisfied $\sum_{i \in K} x_k f_k \leq$

$f^{\max}$. Based on the proportionate allocation described in [20], it can be determined as follows:

$$f_k = \frac{C_k}{\sum\limits_{i \in K} \alpha_k} f^{\max}. \tag{8}$$

The BS remote execution latency for processing sensing data of IoT device $k$ is as

$$T_k^{0,\text{comp}} = \frac{\alpha_k C_k}{f_k}. \tag{9}$$

The latency of the IoT device in completing the execution of its offloaded computational task at BS can be expressed as

$$T_k^{0,\text{exe}} = T_k^{0,\text{ul}} + T_k^{0,\text{comp}}. \tag{10}$$

If the task sensing data is processed by vehicle $n$, the binary variable is denoted $y_k^n \in \{0,1\}$ as a decision variable, where $y_k^n = 1$ if the offloaded sensing task of IoT device $k$ is forwarded from BS to the vehicle $n$ and $y_k^n = 0$, otherwise.

The downlink transmission rate from BS to vehicle $n$ is

$$R_n^{0\to\text{ve}} = B_n log_2 \left(1 + \frac{g_n q_n}{\sigma^2}\right), \tag{11}$$

where $q_n$ is the transmit power assigned by the BS for downlink transmission to the vehicle $n$, $g_n$ is channel gain between the BS server and vehicle $n$, and $B_n$ is the bandwidth between BS and vehicle $n$. The transmission latency from BS to vehicle $n$ can be expressed as

$$T_k^{n,\text{dl}} = \frac{\alpha_k}{R_n^{0\to\text{ve}}}. \tag{12}$$

$f_n^{\text{ve}}$ is the computing capacity of the vehicle $n$, the execution latency on the vehicle $n$ for processing sensing data of IoT device $k$ is given as

$$T_k^{n,\text{comp}} = \frac{\alpha_k C_k}{f_n^{\text{ve}}}. \tag{13}$$

Therefore, when the offloaded task of IoT device $k$ is calculated at the vehicle $n$, the total latency can be expressed as

$$T_k^{n,\text{exe}} = T_k^{0,\text{ul}} + T_k^{n,\text{dl}} + T_k^{n,\text{comp}}. \tag{14}$$

### *E. Problem Formulation*

Let $\boldsymbol{s} = \{s_k\}$ be the sensing rate vector, $\boldsymbol{\alpha} = \{\alpha_k\}$ be the portion of the offloaded sensing task vector, $\boldsymbol{\beta} = \{\beta_k\}$ is the bandwidth resource allocation vector, $\boldsymbol{x} = \{x_k\}$ is a set vector of the computation decision of BS that indicates whether or not offloaded sensing task of IoT device $k$ is performed at BS, and $\boldsymbol{y} = \{y_k^n\}$ is a set of tasks offloading decisions of the vehicles. The objective function aims to minimize the total latency of IoT devices in the system by optimizing the data sensing rate, task offloading, and resource allocation problem as $\boldsymbol{T}(\boldsymbol{s},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y}) = \sum\limits_{k\in\mathcal{K}} \boldsymbol{T}_k(\boldsymbol{s},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})$, where

$$\boldsymbol{T}_k(\boldsymbol{s},\boldsymbol{\alpha},\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{y}) = T_k^{\text{sen}} + T_k^{\text{loc}} + x_k T_k^{0,exe} + \sum_{n\in\mathcal{N}} y_k^n T_k^{n,\text{exe}}. \tag{15}$$

Thus, the optimization problem is given by

$$\mathbf{PA}: \min_{\boldsymbol{s},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y}} \boldsymbol{T}(\boldsymbol{s},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y}) \tag{16a}$$

subject to

$$\text{C1: } 0 < s_k \le S_k^{\max},\ \forall k \in \mathcal{K}, \tag{16b}$$

$$\text{C2: } 0 \le \alpha_k \le D_k,\ \forall k \in \mathcal{K}, \tag{16c}$$

$$\text{C3: } \sum\nolimits_{k\in\mathcal{K}} \beta_k \le 1, \tag{16d}$$

$$\text{C4: } \beta_k \in [0,1],\ \forall k \in \mathcal{K}, \tag{16e}$$

$$\text{C5: } E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} \le E_k^{\max},\ \forall k \in \mathcal{K}, \tag{16f}$$

$$\text{C6: } x_k + \sum\nolimits_{n\in\mathcal{N}} y_k^n = 1,\ \forall k \in \mathcal{K}, \tag{16g}$$

$$\text{C7: } x_k \in \{0,1\},\ \forall k \in \mathcal{K}, \tag{16h}$$

$$\text{C8: } y_k^n \in \{0,1\},\ \forall k \in \mathcal{K}, \forall n \in \mathcal{N}, \tag{16i}$$

where $S_k^{\max}$ represents the maximum data sensing rate, and $E_k^{max}$ is the maximum allowable energy consumption of the IoT device $k$. Constraint (16b) ensures that the sensing rate of IoT devices does not exceed its maximum value. Constraint (16c) states that the portion of the offloaded sensing task of an IoT device must be smaller than the entire sensing data size. Constraints (16d) and (16e) ensure that the total communication resource, particularly the bandwidth, assigned to all IoT devices remains within the bounds of the maximum available bandwidth. Constraint (16f) is the energy consumption requirement of IoT devices. Constraint (16g) ensures that the offloaded sensing task of an IoT device is processed in one location at most, i.e., at the BS or the vehicles. Constraints (16h) and (16i) involve binary decision variables.

## IV. Solution Approaches

The problem is categorized as an MINLP problem, a type that is commonly recognized as NP-hard. These computational difficulties were addressed by decomposing the problem into four feasible subproblems. In this decomposition, the subproblems were transformed into convex problems and were solved alternately. The optimal solution for each subproblem can be obtained because of their convex nature. The proposed approach involves decomposing the problem into four subproblems. The proposed solution framework is depicted in Fig. 2.

### *A. Optimal Data Sensing Rate (ODSR)*

The sub-problem data-sensing rate for a given fraction of offloaded task $\boldsymbol{\alpha}$, bandwidth allocation $\boldsymbol{\beta}$, and computation offloading decisions $\boldsymbol{x}$ and $\boldsymbol{y}$ is formulated as follows:

$$\mathbf{P1}: \min_{\boldsymbol{s}} \boldsymbol{T}(\boldsymbol{s}) \tag{17a}$$

subject to

$$0 < s_k \le S_k^{\max},\ \forall k \in \mathcal{K}, \tag{17b}$$

$$E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} \le E_k^{\max},\ \forall k \in \mathcal{K}, \tag{17c}$$

The problem in (17) is a convex optimization problem.

**Lemma 1:** With given values of $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})$, the first-order derivatives of $\boldsymbol{T}(\boldsymbol{s})$ with respect to (w.r.t.) $s_k$ are taken as

$$\frac{\partial \boldsymbol{T}(\boldsymbol{s})}{\partial s_k} = -\frac{D_k}{s_k^2}, \forall k \in \mathcal{K}. \tag{18}$$

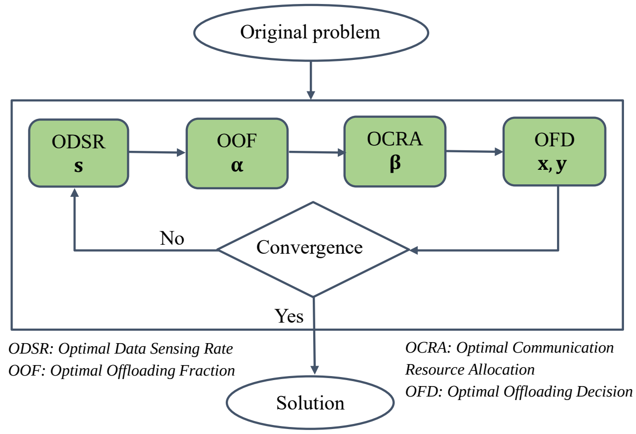


Fig. 2: The proposed solution framework.

The second-order derivatives of $\boldsymbol{T}(\boldsymbol{s})$ w.r.t. $s_k$ are expressed as

$$\frac{\partial^2 \boldsymbol{T}(\boldsymbol{s})}{\partial s_k^2} = \frac{2D_k}{s_k^3} \succ 0, \forall k \in \mathcal{K}. \tag{19}$$

Hence, the objective function in equation (17) is convex. Moreover, the constraints in equation (17) (i.e., C1 and C5) are linear and convex. Therefore, problem (17) is a convex optimization problem. Thus, it can be resolved using the dual problem. The Lagrangian function of problem (17) can be obtained as

$$\begin{aligned}\mathcal{L}(\boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \sum\nolimits_{k\in\mathcal{K}} \frac{D_k}{s_k} + \sum\nolimits_{k\in\mathcal{K}} \lambda_k \left(s_k - S_k^{\max}\right) \\ &+ \sum\nolimits_{k\in\mathcal{K}} \nu_k \left(E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} - E_k^{\max}\right)\end{aligned} \tag{20}$$

where $\lambda \succeq 0$ and $\nu \succeq 0$ are the vectors of the Lagrangian multiplier given by constraints C1 and C5. The dual problem of **P1** is expressed as follows:

$$\max \min_{\boldsymbol{s}} \boldsymbol{L}(\boldsymbol{s}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{21a}$$
$$\text{s.t. } \lambda \succeq 0, \nu \succeq 0.$$

The sub-gradient method can be used to resolve the dual problem. The Lagrangian multipliers are updated as

$$\boldsymbol{\lambda}_k^{(i+1)} = \left[\boldsymbol{\lambda}^{(i)} + \mu_1 \left(s_k - S_k^{\max}\right)\right]^+, \tag{22}$$

$$\boldsymbol{\nu}_k^{(i+1)} = \left[\boldsymbol{\nu}^{(i)} + \mu_2 \left(E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} - E_k^{\max}\right)\right]^+, \tag{23}$$

where $\mu_1$ and $\mu_2$ are positive step sizes. Under the Karush–Kuhn–Tucker (KKT) conditions, the sensing rate variable $\boldsymbol{s}$ in **P1** is updated by solving the following:

$$\frac{\partial \boldsymbol{\mathcal{L}}}{\partial s_k} = -\frac{D_k}{s_k^2} + \lambda_k + 2\nu_k \kappa s_k D_k W_k = 0 \tag{24}$$

Based on the aforementioned study, the optimal data sensing rate (ODSR) algorithm is provided in Algorithm 1 using a Lagrangian relaxation method.

**Algorithm 1** ODSR algorithm

**Initialization:** $i = 0, \lambda_k(0) = 0, \nu_k(0) = 0, s_k(0) = 0, \epsilon > 0$, and fixed step sizes $\mu_1$ and $\mu_2$;
**repeat**
  Increase the iteration index $i = i + 1$;
  Update Lagrangian multipliers $\lambda_k^{(i)}$ and $\nu_k^{(i)}$ according to (22) and (23);
  Update $\beta_k^{(i)}$ by solving (24);
**until** $|s_k^{(i)} - s_k^{(i-1)}| \leq \epsilon$;
Finally, set $s_k^* = s_k^{(i)}$ as the final solution;

### *B. Optimal Offloading Fraction (OOF)*

The sub-problem offloading fraction for a given data sensing rate $\boldsymbol{s}$, bandwidth allocation $\boldsymbol{\beta}$, and computation offloading decisions $\boldsymbol{x}$ and $\boldsymbol{y}$ is formulated as follows:

$$\mathbf{P2}: \min_{\boldsymbol{\alpha}} \ \boldsymbol{T}(\boldsymbol{\alpha}) \tag{25a}$$
$$\text{subject to}$$
$$0 \leq \alpha_k \leq D_k, \ \forall k \in \mathcal{K}, \tag{25b}$$
$$E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} \leq E_k^{\max}, \ \forall k \in \mathcal{K}. \tag{25c}$$

The problem in (25) is convex because of the linear objective function and set constraints. Thus, existing software tools were used, i.e., CVX package, to solve the problem (25).

### *C. Optimal Communication Resource Allocation (OCRA)*

The sub-problem communication resource allocation for a given data sensing rate $\boldsymbol{s}$, fraction of offloaded task $\boldsymbol{\alpha}$, and computation offloading decisions $\boldsymbol{x}$ and $\boldsymbol{y}$ is formulated as

$$\mathbf{P3}: \min_{\boldsymbol{\beta}} \ \boldsymbol{T}(\boldsymbol{\beta}) \tag{26a}$$
$$\text{subject to}$$
$$\sum_{k\in\mathcal{K}} \beta_k \leq 1, \tag{26b}$$
$$\beta_k \in [0,1], \ \forall k \in \mathcal{K}, \tag{26c}$$
$$E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} \leq E_k^{\max}, \ \forall k \in \mathcal{K}, \tag{26d}$$

Problem in (26) is a convex optimization problem.

**Lemma 2:** With a given $(\boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{x}, \boldsymbol{y})$, the first-order derivatives of $\boldsymbol{T}(\boldsymbol{\beta})$ w.r.t. $\beta_k$ may be represented as

$$\frac{\partial \boldsymbol{T}(\boldsymbol{\beta})}{\partial \beta_k} = \frac{-\alpha_k}{\beta_k^2 B^{\text{ul}} \log_2\left(1 + h_k p_k/\sigma^2\right)}, \forall k \in \mathcal{K}. \tag{27}$$

The second-order derivative of $\boldsymbol{T}(\boldsymbol{s})$ w.r.t. $s_k$ is

$$\frac{\partial^2 \boldsymbol{T}(\boldsymbol{\beta})}{\partial \beta_k^2} = \frac{2\alpha_k}{\beta_k^3 B^{\text{ul}} \log_2\left(1 + h_k p_k/\sigma^2\right)}, \forall k \in \mathcal{K}. \tag{28}$$

We can verify that $\partial^2 \boldsymbol{T}(\boldsymbol{\beta})/\partial \beta_k^2 \succ 0$. Hence, the objective function as depicted in (26) is convex. Furthermore, the constraints within C3, C4, and C5 are both linear and convex. Consequently, the optimization problem in (26) can be classified as convex. Therefore, it can be resolved effectively using the dual problem. The Lagrangian function of (26) can be obtained as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta},\boldsymbol{\psi},\boldsymbol{\phi}) &= \sum\nolimits_{k\in\mathcal{K}} \frac{\alpha_k}{\beta_k B^{\text{ul}} \log_2\left(1+h_k p_k/\sigma^2\right)} \\ &+ \psi\left(\sum\nolimits_{k\in\mathcal{K}} \beta_k - 1\right) \\ &+ \sum\nolimits_{k\in\mathcal{K}} \phi_k\left(E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} - E_k^{\max}\right) \end{aligned} \tag{29}$$

where $\psi$, and $\boldsymbol{\phi} = \{\phi_k\}_{k\in\mathcal{K}}$ are the non-negative Lagrangian multiplier given by constraints C3, C4, and C5. The dual problem of **P3** can be expressed as follows:

$$\max \min_{\boldsymbol{\beta}} \boldsymbol{L}(\boldsymbol{\beta},\boldsymbol{\psi},\boldsymbol{\phi}) \tag{30a}$$

$$\text{s.t. } \psi \geq 0, \boldsymbol{\phi} \succeq 0.$$

The sub-gradient method can be used to solve the dual problem. The Lagrangian multipliers are updated as

$$\psi^{(j+1)} = \left[\psi^{(j)} + \omega_1\left(\sum\nolimits_{k\in\mathcal{K}} \beta_k - 1\right)\right]^+, \tag{31}$$

$$\phi_k^{(j+1)} = \left[\phi_k^{(j)} + \omega_2\left(E_k^{\text{sen}} + E_k^{\text{loc}} + E_k^{0,\text{ul}} - E_k^{\max}\right)\right]^+, \tag{32}$$

where $\omega_1$ and $\omega_2$ are positive step sizes. Under the KKT conditions, the optimal solution for bandwidth resource allocation $\boldsymbol{\beta}$ in **P3** is given as

$$\beta_k^* = \sqrt{\frac{\alpha_k(1+\phi_k p_k)}{\psi B^{\text{ul}} \log_2\left(1+h_k p_k/\sigma^2\right)}}. \tag{33}$$

Based on the aforementioned study, the optimal communication resource allocation (OCRA) algorithm is provided in Algorithm 2 using a Lagrangian relaxation method.

**Algorithm 2** OCRA algorithm

1: **Initialization:** $j = 0, \beta_k^{(0)} = 0, \psi^{(0)} = 0, \boldsymbol{\phi}_k^{(0)} = 0, \epsilon > 0,$ and fixed step sizes $\omega_1$ and $\omega_2$;
2: **repeat**
3: Increase the iteration index $j = j + 1$;
4: Update Lagrangian multipliers $\psi^{(j)}$ and $\phi_k^{(j)}$ according to (31) and (32);
5: Update $\beta_k^{(j)}$ according to (33);
6: **until** $|\beta_k^{(j)} - \beta_k{}^{(j-1)}| \leq \epsilon$;
7: Finally, set $\boldsymbol{\beta}_k^* = \boldsymbol{\beta}_k^{(j)}$ as the final solution;

### *D. Optimal Offloading Decision (OFD)*

The sub-problem offloading decision for a given data sensing rate $\boldsymbol{s}$, fraction of offloaded task $\boldsymbol{\alpha}$, and bandwidth allocation $\boldsymbol{\beta}$ can be expressed as

$$\mathbf{P4}: \min_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{T}(\boldsymbol{x},\boldsymbol{y}) \tag{34a}$$

subject to

$$x_k + \sum_{n\in\mathcal{N}} y_k^n = 1,\ \forall k \in \mathcal{K}, \tag{34b}$$

$$x_k \in \{0,1\},\ \forall k \in \mathcal{K}, \tag{34c}$$

$$y_k^n \in \{0,1\},\ \forall k \in \mathcal{K}, \forall n \in \mathcal{N}. \tag{34d}$$

From the optimization problem (34), decision variables are coupled in (34) and C6. Furthermore, the decision variables $\boldsymbol{x},\boldsymbol{y}$ are binary. As a result, (34) is an NP-hard problem. The integer variable was handled by reformatting the optimization problem (34) by relaxing the binary variables into continuous variables and transforming the objective function as follows:

$$\mathbf{P4}: \min_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{T}(\boldsymbol{x},\boldsymbol{y}) \tag{35a}$$

subject to

$$x_k + \sum_{n\in\mathcal{N}} y_k^n = 1,\ \forall k \in \mathcal{K}, \tag{35b}$$

$$x_k \in [0,1],\ \forall k \in \mathcal{K}, \tag{35c}$$

$$y_k^n \in [0,1],\ \forall k \in \mathcal{K}, \forall n \in \mathcal{N}, \tag{35d}$$

The optimization problem in (35) is difficult to solve because of the coupling constraints in (35b). Therefore, the BSUM optimization approach [21] was applied to solve the optimization problem in (35). Owing to its benefits, BSUM has been used to solve various of complex optimization problems for which it can offer a reliable approximation of the solution [2]. The optimization problem was expressed succinctly as follows to make the notation simpler:

$$\min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathbf{Y}} \boldsymbol{T}(\boldsymbol{x},\boldsymbol{y}), \tag{36}$$

where $\mathcal{X} \triangleq \{\boldsymbol{x} : x_k + \sum_{n\in\mathcal{N}} y_k^n = 1,\ \forall k \in \mathcal{K}, x_k \in [0,1],\ \forall k \in \mathcal{K}\}$, and $\mathcal{Y} \triangleq \{\boldsymbol{y} : x_k + \sum_{n\in\mathcal{N}} y_k^n = 1,\ \forall k \in \mathcal{K}, y_k^n \in [0,1],\ \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$ are the feasible sets of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The upper-bound proximal function $T_i$ of the objective function in (36) was identified for each iteration $r$, $\forall r \in \mathcal{B}^r$, where $\mathcal{B}$ is the set of indices. A quadratic penalization was added to the objective function in (36) to ensure that the proximal upper-bound function $T_i$ is convex, as follows:

$$\mathcal{T}_i(\boldsymbol{x}_i;\boldsymbol{x}^{(r)},\boldsymbol{y}^{(r)}) = \mathcal{T}(\boldsymbol{x}_i;\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) + \frac{\rho_i}{2}\|(\boldsymbol{x}_i - \tilde{\boldsymbol{x}})\|^2, \tag{37}$$

where $\rho_i > 0$ is the positive penalty parameter. The quadratic penalty term makes the problem in (36) strictly convex. The variable $y$ can also be included with the approximate function. In other words, the approximation function returns distinct minimizer vectors $\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}$ with respect to $\boldsymbol{x}$, and $\boldsymbol{y}$ for each iteration $r$, which is regarded as the answer to the preceding step of $(r-1)$. The following sub-problems can be solved to alter the solution for each iteration $(t+1)$:

$$\boldsymbol{x}_i^{(r+1)} = \operatorname*{argmin}_{\boldsymbol{x}_i\in\mathcal{X}} \mathcal{T}_i(\boldsymbol{x}_i;\boldsymbol{x}^{(r)},\boldsymbol{y}^{(r)}), \tag{38}$$

$$\boldsymbol{y}_i^{(r+1)} = \operatorname*{argmin}_{\boldsymbol{y}_i\in\mathcal{Y}} \mathcal{T}_i(\boldsymbol{y}_i;\boldsymbol{y}^{(r)},\boldsymbol{x}^{(r+1)}). \tag{39}$$

We suggested BSUM approach can be used to resolve the sub-problems in (38) and (39). The detail of the BSUM-based offloading decision algorithm is stated in Algorithm 3.

### *E. Complexity Analysis*

In this section, we analyze the complexity of the proposed algorithm. We first use the Lagrangian relaxation-based algorithm to solve P1. The Lagrangian multipliers, i.e., $\lambda$ and $\nu$,

**Algorithm 3** BSUM-based offloading decision algorithm

1: **Initialization:** Set the iteration index $r = 0$, find initial feasible solutions $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)})$, and $\epsilon > 0$;
2: **repeat**
3: Increase the iteration index $r = r + 1$;
4: Select index set $\mathcal{B}^r$.
5: Let $\boldsymbol{x}_i^{(r)} = \underset{\boldsymbol{x}_i \in \mathcal{X}}{\text{argmin}}\ \mathcal{T}_i(\boldsymbol{x}_i; \boldsymbol{x}^{(r-1)}, \boldsymbol{d}^{(r-1)})$;
6: Set $\boldsymbol{x}_j^{(r)} = \boldsymbol{x}_j^{(r)}, \forall j \notin \mathcal{B}^r$;
7: Similarly, solve the problem in (39) to obtain $\boldsymbol{y}_i^{(r)}$;
8: **until** $\|\frac{\mathcal{T}_i^{(r)} - \mathcal{T}_i^{(r-1)}}{\mathcal{T}_i^{(r-1)}}\| \le \epsilon$;
9: Finally, set $(\boldsymbol{x}_i^{(r)}, \boldsymbol{y}_i^{(r)})$ as the final solution;

are updated via the sub-gradient method to find the optimal data sensing rate in Algorithm 1. Precisely, $K$ sensing rate prices and $K$ energy prices of IoT devices are updated in each iteration as expressed in (22) and (23). Thus, the computational complexity of the sub-gradient method is $\mathcal{O}(K^4)$ [22]. Then, to solve convex problem P2, we use the CVX toolkit, in which the interior point method is implemented. Thus, to solve the problem with $K$ variables, the computational complexity of the CVX toolkit is $\mathcal{O}(K^{3.5})$. Furthermore, we use the Lagrangian relaxation algorithm to find the solution to the optimal resource allocation problem, as shown in Algorithm 3. Thus, the complexity of Algorithm 3 is $\mathcal{O}(K^2)$. Additionally, we use the CVX toolkit to solve the sub-problems in (38) and (39). As a result, the computational complexity of the BSUM algorithm expressed in Algorithm 3 is $\mathcal{O}(K^{3.5}) + \mathcal{O}(N^{3.5}K^{3.5})$. Therefore, the overall computational complexity of our proposed optimization framework illustrated in Fig. 2 is $\mathcal{O}(\hat{J}(K^4 + K^{3.5} + K^2 + K^{3.5} + N^{3.5}K^{3.5}))$, where $\hat{J}$ is the number of iterations.

## V. Performance Evaluation

This section presents the simulation results to examine the performance of the proposed algorithm and compares it with other schemes. The benefits of the proposed approach were compared with three benchmark schemes for decreasing the total latency of IoT devices. The three benchmark schemes are described below.

- *Local only:* In this approach, all IoT devices perform their sensing tasks locally without any offloading (i.e., $\alpha_k = 0$);
- *Uniform offloading:* In this approach, resources are equally allocated for each task by dividing the total capacity of the bandwidth resource by the number of IoT devices. Each sensing task is divided equally between local execution and offloading to the BS;
- *MEC only:* In this approach, only the BS is used for offloading. There is no vehicle assistance in the offloading process, and all sensing tasks are processed locally or at the BS.

### A. Simulation Settings

The simulation parameters were as follows. A system configuration with a BS installed with an MEC server in the middle of a $200 \times 200$ m$^2$ area had $K = 20$ IoT devices and $N = 5$ vehicles. The total bandwidth for uplink from the IoT devices to the BS was $B^{\text{ul}} = 20$ MHz. The available bandwidth for the vehicle $n$ connection with the BS is $B_n = 8$ MHz. The noise power was set to $\sigma^2 = -174$ dBm/Hz. In this scenario, the uplink transmit power of each IoT device to transfer the offloaded sensing task to the BS was fixed at $p_k = 23$ dBm, and the downlink transmit power of the BS to each vehicle was set to $q_n = 30$ dBm. For the IoT device, the number of CPU cycles for sensing and processing one-bit data was distributed randomly with $W_n \in [10, 20]$ cycles and $C_n \in [30, 40]$ cycles, respectively. The sensing data size of the IoT device $k$ was distributed randomly with $D_k \in [60, 80]$ Mb. Furthermore, each IoT device has a maximum allowable energy consumption limit set at $E_k^{max} = 5$J, and the maximum sensing rate for each IoT device is $s_k = 10^8$ bits/s. The local computing capacity of the IoT devices and vehicles was distributed uniformly by $f_n^{\text{loc}} \in [0.5, 0.7]$ GHz and $f_n^{\text{ve}} \in [4, 5]$ GHz, respectively. The maximum computing resource of the MEC server was $f^{\max} = 30$ GHz.

### B. Simulation Results

Fig. 3 illustrates the convergence of the proposed algorithm. The algorithm converged to the stationary point after a relatively small number of iterations, highlighting its effectiveness in reaching the final solution. This demonstrates the efficacy of the proposed approach.

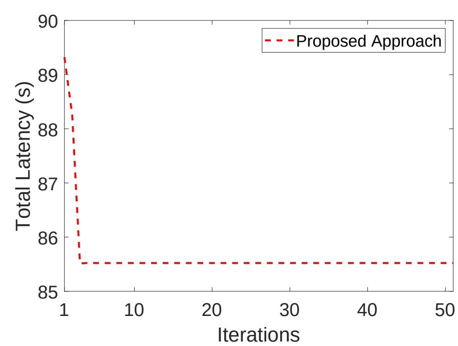


Fig. 3: Convergence of the proposed algorithm.

Fig. 4 shows the total latency versus the quantity of IoT devices across different strategies. This visual representation highlights how the augmentation of IoT devices corresponds to increased latency across all schemes because the relationship between the total latency of the IoT devices and their quantity is linear. Nevertheless, the proposed approach achieves the lowest latency. This is because the proposed approach optimizes the joint problems of sensing rate, offloading decisions, and resource allocation with the assistance of vehicles to reduce the computational pressure at the MEC server. The *uniform offloading* approach has the highest latency because the offloaded sensing data and bandwidth resource allocation are not optimized.

Fig. 5 compares the total energy consumption versus the number of IoT devices. The proposed approach exhibited the lowest energy consumption compared to other schemes. The

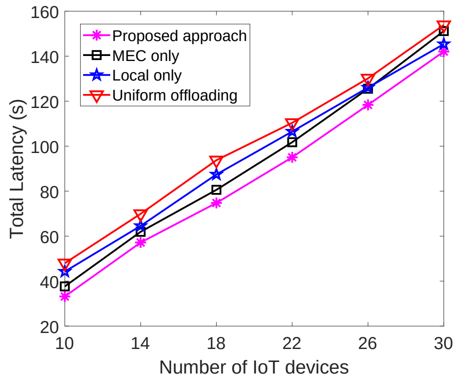


Fig. 4: Comparison of the total latency versus the number of IoT devices.

*Local only* scheme consumes the most energy because the IoT device needs to utilize it for sensing and computation tasks. According to Fig. 4 and Fig. 5, the *Local only* scheme is beneficial regarding latency but requires the IoT device to consume more energy.

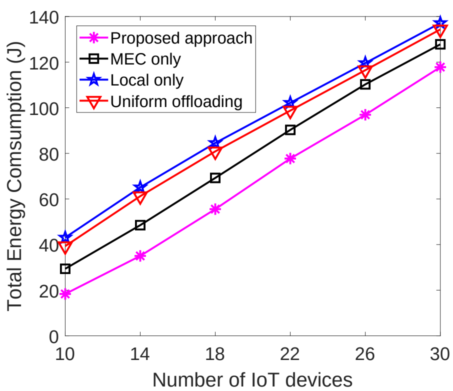


Fig. 5: Comparison of the total energy consumption in relation to the quantity of IoT devices.

The total latency regarding the sensing data size was investigated. The proposed strategy outperformed the alternative approaches, as shown in Fig. 6. The total latency of all approaches increased as the augmentation of sensing data size increased. Among the various schemes, the *uniform offloading* approach exhibited the highest latency. This is because the sensing data are distributed equally between local execution and offloading to the base station.

Fig. 7 shows the total latency experienced by IoT devices in relation to the computational capacity of the MEC server. In the *Local only* scheme, the total latency remained constant regardless of the increasing computational capability of the MEC server. This consistency arose because all IoT devices perform their sensing tasks locally. Consequently, augmenting the capacity of the MEC server does not affect the overall latency of IoT devices. As the computing capacity of the MEC increased, the total latency of the three schemes (i.e., *MEC only*, *uniform offloading*, and the proposed approach) decreased because of the availability of more computing resources. Fig. 7 shows that the proposed approach outperformed *Local only*, *uniform offloading*, and *MEC only*.

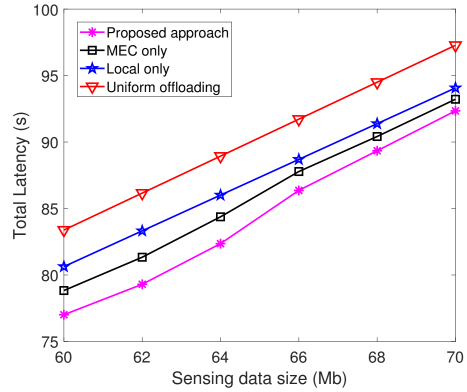


Fig. 6: Comparison of the total latency versus the sensing data size.

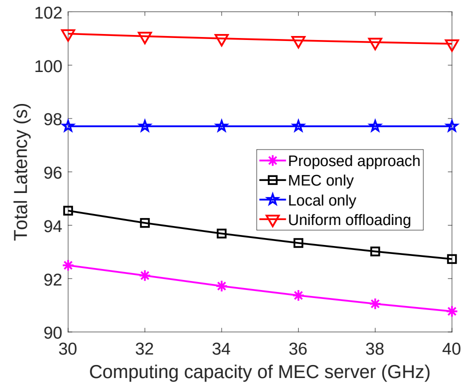


Fig. 7: Comparison of the total latency versus computing capacity of the MEC server.

## VI. Conclusion

This study examined data sensing and computation offloading in a vehicle-assisted MEC system. By leveraging vehicle resources, the latency of IoT devices was minimized by optimizing a joint optimization problem that encompasses sensing rate, offloading decision, partial offloading, and resource allocation. An algorithm that ensures convergence to the optimal solution was introduced to address this challenge. The simulation results validated the effectiveness of the proposed solution, which outperformed various benchmark schemes. Future studies will integrate terrestrial and aerial communications into the system to enhance its capabilities.

## References

[1] Y. Chen, W. Gu, J. Xu, Y. Zhang, and G. Min, "Dynamic task offloading for digital twin-empowered mobile edge computing via deep reinforcement learning," *China Communications*, 2023.

[2] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359–1374, Mar. 2020.

[3] B. Zhu, K. Chi, J. Liu, K. Yu, and S. Mumtaz, "Efficient offloading for minimizing task computation delay of noma-based multiaccess edge computing," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3186–3203, 2022.

[4] S. Chakraborty and K. Mazumdar, "Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1552–1568, 2022.

[5] Z. Zhou, X. Li, C. You, K. Huang, and Y. Gong, "Joint sensing and communication rates control for energy efficient mobile crowd sensing," vol. 22, no. 2, pp. 1314–1327, Feb. 2023.

[6] Z. Liang, H. Chen, Y. Liu, and F. Chen, "Data sensing and offloading in edge computing networks: TDMA or NOMA?" *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4497–4508, Jun. 2022.

[7] X. Li, S. Bi, Z. Quan, and H. Wang, "Online cognitive data sensing and processing optimization in energy-harvesting edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6611–6626, 2022.

[8] X. Li, C. You, S. Andreev, Y. Gong, and K. Huang, "Wirelessly powered crowd sensing: Joint power transfer, sensing, compression, and transmission," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 391–406, 2018.

[9] X. Huang, P. Li, and R. Yu, "Social welfare maximization in container-based task scheduling for parked vehicle edge computing," *IEEE Communications Letters*, vol. 23, no. 8, pp. 1347–1351, 2019.

[10] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications," *IEEE Access*, vol. 6, pp. 66 649–66 663, 2018.

[11] Z. Aghapour, S. Sharifian, and H. Taheri, "Task offloading and resource allocation algorithm based on deep reinforcement learning for distributed AI execution tasks in IoT edge computing environments," *Computer Networks*, vol. 223, p. 109577, 2023.

[12] S. Chen, X. Ge, Q. Wang, Y. Miao, and X. Ruan, "DDPG-based intelligent rechargeable fog computation offloading for IoT," *Wireless NetworFks*, vol. 28, no. 7, pp. 3293–3304, 2022.

[13] K. Wang, Z. Hu, Q. Ai, Y. Zhong, J. Yu, P. Zhou, L. Chen, and H. Shin, "Joint offloading and charge cost minimization in mobile edge computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 205–216, Feb 2020.

[14] C. Shang, Y. Sun, H. Luo, and M. Guizani, "Computation offloading and resource allocation in NOMA-MEC: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, 2023.

[15] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 392–407, Jun 2019.

[16] C. Ma, J. Zhu, M. Liu, H. Zhao, N. Liu, and X. Zou, "Parking edge computing: parked-vehicle-assisted task offloading for urban vanets," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9344–9358, 2021.

[17] A. Zhou, X. Ma, S. Gao, and S. Wang, "Providing reliable service for parked-vehicle-assisted mobile edge computing," *ACM Transactions on Internet Technology*, vol. 22, no. 4, pp. 1–24, 2022.

[18] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.

[19] X.-Q. Pham, T. Huynh-The, E.-N. Huh, and D.-S. Kim, "Partial computation offloading in parked vehicle-assisted multi-access edge computing: A game-theoretic approach," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 10 220–10 225, 2022.

[20] Y. K. Tun, N. H. Tran, D. T. Ngo, S. R. Pandey, Z. Han, and C. S. Hong, "Wireless network slicing: Generalized kelly mechanism-based resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1794–1807, 2019.

[21] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, Dec. 2016.

[22] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.