

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Joint Wireless Resource and Computation Offloading Optimization for Energy Efficient Internet of Vehicles

Dimitrios Pliatsios, Panagiotis Sarigiannidis, Thomas Lagkas, Vasileios Argyriou, Alexandros-Apostolos A. Boulogeorgos, Peristera Baziana

Abstract—The Internet of Vehicles (IoV) is an emerging paradigm, which is expected to be an integral component of beyond-fifth-generation and sixth-generation mobile networks. However, the processing requirements and strict delay constraints of IoV applications pose a challenge to vehicle processing units. To this end, multi-access edge computing (MEC) can leverage the availability of computing resources at the edge of the network to meet the intensive computation demands. Nevertheless, the optimal allocation of computing resources is challenging due to the various parameters, such as the number of vehicles, the available resources, and the particular requirements of each task. In this work, we consider a network consisting of multiple vehicles connected to MEC-enabled roadside units (RSUs) and propose an approach that minimizes the total energy consumption of the system by jointly optimizing the task offloading decision, the allocation of power and bandwidth, and the assignment of tasks to MEC-enabled RSUs. Due to the original problem complexity, we decouple it into subproblems and we leverage the block coordinate descent method to iteratively optimize them. Finally, the numerical results demonstrate that the proposed solution can effectively minimize total energy consumption for various numbers of vehicles and MEC nodes while maintaining a low outage probability.

Index Terms—6G, B5G, Block Coordinate Descent, Computation Offloading, Energy Efficiency, Internet of Vehicles, Mobile Edge Computing

I. INTRODUCTION

The sixth-generation (6G) of mobile networks aims to integrate the advances in wireless communication technologies to deliver enhanced performance compared to fifth-generation (5G) mobile networks and realize new applications and services [1], requiring increased computing capabilities and low computing latency. The Internet of Vehicles (IoV) is an emerging paradigm derived from the concept of the Internet of Things and features great potential in the Beyond

5G/6G era [2]–[5]. The IoV paradigm aims to deliver an intelligent and efficient transportation system able to support applications such as autonomous driving, traffic prediction, and road security and safety [6]. Such applications often have strict delay constraints and require intensive computations [7]. Although the computing capabilities of vehicles are higher than conventional mobile devices, the complex processing requirements and strict delay constraints of IoV applications pose a challenge to vehicle processing units. In addition, an individual vehicle’s available computing resources may not be able to meet the aforementioned requirements and constraints.

Multi-access edge computing (MEC), formerly known as mobile edge computing, can leverage the availability of computing resources located at the edge of the network to efficiently realize computing resource sharing, in order to meet the intensive computing demands posed by IoV applications. In this direction, a device can offload a task to a MEC-enabled small cell (SC), where sufficient computation resources exist. Nevertheless, the orchestration of resource sharing among various devices and SCs is challenging due to the heterogeneity of the resources and the time-varying topology of vehicular networks. Furthermore, the dense deployment of MEC-enabled SCs will result in higher total energy consumption. Consequently, minimizing the total energy consumption while taking into account the quality of service (QoS) requirements of the application, is challenging [8], [9].

A. Related Works

In this direction, research efforts are being focused on exploiting the ample computing resources of the edge nodes by offloading the tasks of mobile devices or vehicles. In more detail, the minimization of the task processing time is the focus of the research works in [10]–[19]. The authors in [10] developed a task offloading optimization approach that aims to minimize task computation delay and energy consumption. Xu *et al.* [11] investigated an offloading system, where the QoS depends on the task response time and developed a deep reinforcement learning approach that minimizes the response time. Zhao *et al.* [12] considered the partial offloading of vehicle tasks to multiple smart devices, such as drones and edge nodes, and minimized the execution time of a task taking into account energy consumption and rental rate of the smart device. In [13], the authors combined reinforcement learning and heuristic algorithms to optimize the allocation of user

This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 957406 (TERMINET).

D. Pliatsios and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani 50100, Greece; {dpliatsios, psarigiannidis}@uowm.gr

T. Lagkas is with the Department of Computer Science, School of Science, International Hellenic University, Thessaloniki, Greece; tlagkas@cs.ihu.gr

V. Argyriou is with the Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, UK; vasileios.argyriou@kingston.ac.uk

A-A.A. Boulogeorgos is with the Department of Digital Systems, University of Piraeus, 18534 Piraeus, Greece; al.boulogeorgos@ieee.org

P. Baziana is with the Department of Computer Science and Telecommunications, University of Thessaly, Lamia, Greece; baziana@uth.gr

applications to vehicular computation resources. Moreover, Luo *et al.* [14] designed a dynamic programming-based algorithm that minimizes the processing latency of tasks in heterogeneous MEC environments. In [15], the authors leveraged reinforcement learning to develop a mobile offloading method aiming to minimize the cost of task migration under energy constraints. The authors of [16] proposed a federated learning-based offloading scheme for minimizing the total latency in vehicular environments, where each task can be divided into three parts so it can be respectively processed locally, offloaded to another vehicle, or offloaded to a MEC node. Yadav *et al.* [17] developed an algorithm to minimize the task latency by optimally selecting the tasks to be offloaded to the MEC nodes. In [18], the authors formulated the minimization of task latency by jointly optimizing the offloading decision, as well as the wireless and computing resource allocation in satellite-assisted vehicle-to-vehicle communications. The authors of [19] utilized the particle swarm optimization algorithm to minimize the processing time of each task by offloading portions of the task to multiple vehicles.

Alternatively, the maximization of the throughput is the focus of the research works presented in [20]–[22]. The authors in [20] investigated the resource allocation in networks consisting of unmanned aerial vehicles and formulated a mixed-integer non-linear problem, aiming to maximize the average throughput while satisfying energy constraints. In addition, Ning *et al.* [21] leveraged non-orthogonal multiple access and MEC technologies to develop a method that maximizes the link throughput, by optimizing power allocation, subchannel assignment, and task assignment. Furthermore, Lu *et al.* [22] considered a network, where two unmanned aerial vehicles (UAVs) provide wireless power transfer to two ground devices and developed a solution based on successive convex programming in order to maximize the sum average transmission rate.

The approaches presented in [23]–[29] are focused on maximizing the system utility. Specifically, Dai *et al.* [23] proposed a low-complexity algorithm to jointly optimize the offloading decision and resource allocation toward maximizing the system utility. In [24], the authors proposed a vehicle-assisted offloading scheme that aims to maximize the long-term utility rate of a vehicular network using a reinforcement learning method. The authors in [25] addressed the maximization of the system offloading utility rate taking into account the task execution order and the available computing resources. The authors of [26] designed a collaborative resource allocation and offloading decision optimization scheme for maximizing the utility rate of the system. In [27], Zhang *et al.* adopted a deep Q-learning method for optimizing the offloading decision and the data uploading method (i.e., vehicle-to-vehicle, vehicle-to-base-station) with the aim of maximizing the system utility rate. The authors in [28] presented a joint resource allocation and task scheduling scheme for maximizing the system's utility by formulating the corresponding optimization problem as a Stackelberg game. Xu *et al.* [29] leveraged a multi-objective evolutionary algorithm based on decomposition to minimize the task processing latency and maximize the utilization of the system resources.

Finally, the research works in [30]–[40] are focused on

minimizing the system energy consumption through the optimal allocation of the available resources. Particularly, the authors in [30] formulated the computation offloading as a mixed-integer non-linear programming problem and proposed a genetic algorithm that minimizes the energy consumption. In [31], the authors leveraged a deep reinforcement learning approach for minimizing the energy consumption through the joint optimization of the offloading decision and the assignment of tasks to the MEC nodes. The authors in [32] investigated the trade-off between the task latency and energy consumption and developed an approach to find the optimal task offloading decision and the allocation of wireless resources. Zhou *et al.* [33] developed a scheme based on the alternating direction method of multipliers for minimizing the total energy consumption of the system by finding the optimal offloading decision for each task. In [34], the authors presented a method based on the Lagrange dual decomposition method for minimizing the energy consumption through the joint optimization of the offloading decision, the allocation of transmission power, and the scaling of computing resources. The authors in [35] proposed an approach that maximizes the system energy efficiency by optimally allocating the offloading transmission power and time, as well as scaling the device chip computing frequency. Jang *et al.* [36] investigated the energy consumption assuming partial and complete offloading in vehicular edge computing environments and proposed a solution for optimally assigning the offloading of the task in time-slots. The authors in [37] developed an energy-efficient fog computation offloading scheme in order to meet the stringent requirements of the industrial internet of things. The scheme leverages an accelerated gradient descent algorithm that optimizes the offloading ratio, the transmission power and time, and the local central processing unit (CPU) computation speed. Wang *et al.* [38] focused on the energy consumption of an edge system and proposed an imitation learning-enabled scheduling algorithm that takes into account the latency constraints of the tasks. In [39], the authors presented a deep reinforcement learning method to minimize the long-term energy consumption and task processing latency through the optimization of the offloading decision and the allocation of computing resources. Lagkas *et al.* [40] developed a joint allocation scheme, involving three optimization phases for the edge, radio, and optical resources, respectively.

B. Contributions

The aforementioned works presented some interesting results, however, some of them are focused on optimizing only a particular aspect of the offloading process (e.g., the offloading decision), while most of them are focused on jointly optimizing the offloading decision and the allocation of the transmission power. Furthermore, some of the research works are focused on the joint optimization of the wireless and computing resources. Of note, the solutions presented in most of the works are based on deep learning or reinforcement learning algorithms to optimize the long-term system performance. However, these algorithms are considered computationally expensive. Moreover, deep learning algorithms require large datasets volumes to achieve high performance.

Motivated by these remarks, we develop a solution that aims to minimize the total energy consumption of the system by optimally offloading the tasks to the MEC-enabled roadside units (RSUs), taking into account the latency requirements and the availability of wireless and computing resources. In particular, the solution aims to jointly optimize the task offloading decision, the allocation of power and bandwidth resources, the assignment of tasks to MEC-enabled RSUs, and the frequency scaling of MEC-enabled RSUs. In more detail, the contributions of this work are as follows:

- We present a scenario consisting of multiple vehicles that are served by a number of RSUs. In the considered scenario, each vehicle may choose to compute its task locally or offload a portion of it in a MEC-enabled RSU. Additionally, the scenario supports task migration, meaning that a task can be migrated from a RSU to another, based on the computation requirements and availability of resources.
- We formulate the minimization of the total energy consumption as a joint optimization of the task offloading decision, the allocation of power and bandwidth, the assignment of tasks to MEC-enabled RSUs, and the frequency scaling of MEC-enabled RSUs. We also discuss the convexity of the original optimization problem and transform it into convex equivalents.
- As the joint optimization problem is challenging to solve, we decouple the original optimization problem into three problems and solve each one in an iterative way by leveraging the block coordinate descent (BCD) method.
- Particularly, for optimizing the task offloading decision, we derive closed-form expressions taking into account each task's latency constraints. Towards optimizing the power and bandwidth allocation, as well as the task assignment and frequency scaling, the Lagrange multipliers and subgradient methods are employed.
- We evaluate the performance of the proposed approach through system-level Monte Carlo simulations in terms of total energy consumption and outage probability.
- To highlight the impact of the allocation of wireless and computing resources on the total energy consumption, we designed three evaluation scenarios. Particularly, in the first scenario, only the offloading decision is optimized, whereas in the second scenario we optimize the offloading decision and the allocation of wireless resources. Finally, in the third scenario, the allocation of both wireless and computing resources is optimized in addition to the offloading decision.

The remainder of the paper is structured as follows: In Section II we develop the model and the problem formulation, while, in Section III, we present the proposed solution. We provide the evaluation results in Section IV and we conclude the work in Section V. Additionally, all notations used throughout the paper are summarized in Table I.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Fig. 1 depicts the considered system model. In particular, a number of vehicles are served by RSUs equipped with MEC

TABLE I: Summary of Notations

Notations	Explanation
N	Number of vehicles
S	Number of RSUs
x_n	Offloading decision
w_n	Bandwidth allocation coefficient
W	Available bandwidth
$a_{n,s}$	Task assignment index
$f_{n,s}$	MEC frequency scaling coefficient
F_s^{max}	Maximum computation capability of MEC node
R_n	Wireless link capacity
B_n	Total system bandwidth
p_n	Vehicle transmission power
$d_{n,s}^{-\delta}$	Distance-based pathloss
σ^2	Noise variance
L_n	Task size
C_n	Required cycles
T_n^{max}	Maximum latency
f_n^{loc}	Local computing capability
T_n^{loc}	Local computation time
T_n^{up}	Upload time
T_n^{off}	Offloaded computation time
E_n^{loc}	Energy consumed locally
E_n^{up}	Energy consumed during the upload
T_n^{mec}	MEC processing time
ϕ_n^{loc}	Local energy consumption coefficient
ϕ_s^{mec}	MEC energy consumption coefficient
$\beta_n, \kappa_{n,s}, \lambda_n, \mu_s$	Lagrange multipliers
$\xi_{n,s}, \pi_n, \tau_s$	Lagrange multipliers

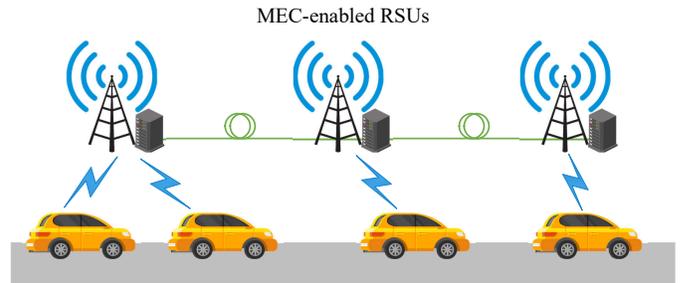


Fig. 1: Computation Offloading for Internet of Vehicles

capabilities. Each vehicle is served by its nearest RSU via a wireless link, while RSUs are interconnected using high-capacity optical backhaul links [41]. The wireless communication between the RSUs and the vehicles can be enabled by a mobile network (e.g., 5G or 6G), while the optical backhaul links can be enabled by the latest optical communications standards, such as the 10-Gigabit Symmetrical Passive Optical Network (10GS-PON) or the Next-Generation PON 2 (NG-PON2), that are able to provide data rates up to 10 Gbps [42], [43].

Let $\mathcal{N} = \{1, \dots, N\}$ denote the set of vehicles, while $\mathcal{S} = \{1, \dots, S\}$ denotes the set of RSUs. To mitigate the energy required for the wireless data transmission, each vehicle is assumed to be connected to the closest RSU in its proximity and the corresponding distance is denoted by $d_{n,s}$. We assume

that the optimization process takes place in cycles, in which the vehicles may offload a portion of a task. Consequently, the terms vehicle and task can be used interchangeably. In addition, for the duration of the cycle, the vehicle position is assumed to remain steady. The offloaded tasks can be part of various IoV applications, including navigation assistance, image or video recognition, collision or obstacle detection, or autonomous driving [44], [45]. In addition, task profilers can be leveraged to provide valuable insights to operators about the computing and delay requirements of each task [46], [47].

A. Communication Model

The wireless link capacity between a vehicle and an RSU is calculated by

$$R_n = w_n W \log_2(1 + \text{SNR}_{n,s}) \quad (1)$$

where w_n denotes the bandwidth portion allocated to vehicle n , while W is the total available bandwidth. The respective signal-to-noise ratio (SNR) is obtained by

$$\text{SNR}_{n,s} = \frac{p_n d_{n,s}^{-\delta}}{\sigma^2} \quad (2)$$

where p_n is the power of the transmitted signal, $d_{n,s}^{-\delta}$ is distance pathloss based on δ coefficient, and σ^2 is the noise variance. Orthogonal frequency-division multiple access (OFDMA) is selected for minimizing interferences among vehicles.

B. Computation Model

Each task n is described by the tuple (L_n, C_n, T_n^{max}) , where L_n denotes the data length in bits to be processed and C_n (cycles/bit) denotes the number of cycles required to process a single bit of the task [21], [48]. Consequently, the total number of cycles required for processing the task can be obtained by $L_n C_n$. Also, T_n^{max} denotes the maximum tolerable latency for the task.

1) *Local Computation*: The total time for the local computation is obtained by

$$T_n^{loc} = \frac{L_n C_n}{f_n^{loc}} \quad (3)$$

where f_n^{loc} (cycles/s) denotes the computing capability of the n -th vehicle. As in [48], [49], and [50], we model the energy consumption of the processor as $\phi_n^{loc} (f_n^{loc})^3$ (joules per second), where ϕ_n^{loc} stands for the processor's chip energy coefficient [49]. By multiplying the aforementioned equation with the right hand side of (3), we obtain the energy consumed for the processing of the n -th task as

$$E_n^{loc} = L_n C_n (f_n^{loc})^2 \phi_n^{loc} \quad (4)$$

2) *Offloaded Computation*: The total time for the offloaded computation consists of the time required for the vehicle to upload the data to the nearest MEC-enabled RSU and the time required for the RSU to process the data. Moreover, the nearest RSU may not have enough available computing resources and thus, the task will be migrated to another RSU through the backhaul optical link. Also, since that the size

of the offloaded task is much smaller than the backhaul link capacity, we can assume that the task migration time is zero in order to simplify the optimization process. To indicate where each task is processed, we use the binary variable $a_{n,s}$ as follows:

$$a_{n,s} = \begin{cases} 1, & \text{the } n\text{-th task is processed at the } s\text{-th node} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Based on the aforementioned remarks, the upload time is calculated as

$$T_n^{up} = \frac{L_n}{R_n} \quad (6)$$

The processing time of the n -th task at the s -th node can be obtained by

$$T_{n,s}^{proc} = \frac{L_n C_n}{f_{n,s} F_s^{max}} \quad (7)$$

where $f_{n,s}$ is the frequency scaling coefficient that denotes the utilization ratio of the processor. For example, when $f_{n,s} = 1$, the current processor frequency will be equal to F_s^{max} , where F_s^{max} denotes the maximum computing capability of the s -th RSU (in Hz).

Assuming that the downlink transmission delay is considered negligible, due to the result of the computation being very small ([51], [52]), the total time for the offloaded computation of the n -th task is

$$T_n^{off} = T_n^{up} + \sum_{s=1}^S a_{n,s} T_{n,s}^{proc} \quad (8)$$

The total energy consumed in the offloaded computation includes the energy consumed at the vehicle for the task upload and the energy consumed at the RSU for the processing. In particular, the energy consumed for the task upload will be the transmission power of the n -th vehicle multiplied by the time required to upload the task and can be calculated by

$$E_n^{up} = p_n T_n^{up} \quad (9)$$

Using the same energy consumption model as in local computation, the energy consumed for the processing of n -th task at the s -th node is obtained by

$$E_{n,s}^{proc} = L_n C_n (f_{n,s} F_s^{max})^2 \phi_s^{mec} \quad (10)$$

where ϕ_s^{mec} is the energy consumption coefficient of the RSU.

$$E_n^{off} = E_n^{up} + \sum_{s=1}^S a_{n,s} E_{n,s}^{proc} \quad (11)$$

For the communication between RSUs, a high-capacity passive optical network is utilized [53], [54]. As a result, in case of task migration, the respective data can be promptly transferred among RSUs, with minimal delay, leading to a small energy overhead.

C. Problem Formulation

We aim to minimize the total power consumption of the system by jointly optimizing the task offloading decision, the allocation of power and bandwidth, the assignment of tasks to MEC-enabled RSUs, and the frequency scaling of MEC-enabled RSUs. Moreover, we adopt a partial offloading scheme, meaning that a task can be concurrently computed locally and in a MEC-enabled RSU. The portion of local and offloaded computation is denoted by x_n . Specifically, when $x_n = 0$ the whole task is computed locally at a vehicle, whereas when $x_n = 1$, the whole task is offloaded to a MEC-enabled RSU. Combining (1) - (11), the total computation time is expressed as

$$T_n(x_n, p_n, w_n, a_{n,s}, f_{n,s}) = (1 - x_n)T_n^{loc} + (x_n)T_n^{off} \quad (12)$$

Similarly, the total energy consumption is formulated as

$$E_n(x_n, p_n, w_n, a_{n,s}, f_{n,s}) = (1 - x_n)E_n^{loc} + (x_n)E_n^{off} \quad (13)$$

Consequently, the optimization problem is expressed as follows:

$$\mathcal{P}0: \min_{\mathbf{x}, \mathbf{p}, \mathbf{w}, \mathbf{a}, \mathbf{f}} \sum_{n=1}^N E_n(x_n, p_n, w_n, a_{n,s}, f_{n,s}) \quad (14a)$$

subject to:

$$\max\{T_n^{off}, T_n^{loc}\} \leq T_n^{max}, \forall n \quad (14b)$$

$$0 \leq x_n \leq 1, \forall n \quad (14c)$$

$$0 \leq p_n \leq P_n^{max}, \forall n \quad (14d)$$

$$0 \leq w_n \leq 1, \forall n \quad (14e)$$

$$\sum_{n=1}^N w_n \leq 1 \quad (14f)$$

$$a_{n,s} = \{0, 1\}, \forall n, s \quad (14g)$$

$$\sum_{n=1}^N a_{n,s} \leq 2, \forall s \quad (14h)$$

$$0 \leq f_{n,s} \leq 1, \forall n, s \quad (14i)$$

$$\sum_{n=1}^N f_{n,s} \leq 1, \forall s \quad (14j)$$

In $\mathcal{P}0$, \mathbf{x} denotes the vector of the task offloading decision, while \mathbf{p} and \mathbf{w} denote the vectors of the transmission power and bandwidth allocation, respectively. Furthermore, \mathbf{a} denotes the task-MEC assignment vector, while \mathbf{f} denotes the frequency scaling coefficient vector. Constraint (14b) enforces that the total computation time of the task does not exceed the maximum tolerable delay. Additionally, constraint (14c) enforces the task offloading portion in the range $[0, 1]$, while (14d) enforces the transmission power between 0 and the P_n^{max} . Similarly, (14e) and (14f) are employed to limit the bandwidth coefficient up to 1. Furthermore, (14g) enforces binary values to $a_{n,s}$, while (14h) limits the tasks computed in a single RSU up to two. Finally, (14i) and (14j) are imposed to limit frequency scaling coefficient up to 1.

In $\mathcal{P}0$, the objective function and constraint (14b) are non-linear due to the logarithm in (1). Moreover, there are product

relationships between the optimization variables in the objective function. For example, for the offloaded computation case, $x_n, p_n, f_{n,s}, a_{n,s}$ are multiplied based on (11). Additionally, (14b) and (14g) make the feasible set non-convex. Therefore, $\mathcal{P}0$ is a non-convex mixed-integer non-linear problem.

III. PROPOSED SOLUTION

This section presents the solution to the formulated optimization problem. In this direction, the original problem is decoupled into three problems, which are iteratively optimized through the BCD method. Particularly, closed-form expressions are derived for solving the task offloading decision. Moreover, the Lagrange multipliers and subgradient methods are employed for solving the wireless and computing resource allocation problems.

A. Optimizing Offloading Decision while Fixing the Rest Optimization Variables

In $\mathcal{P}0$, constraint (14b) makes the feasible set non-convex. Therefore, to transform the feasible set into a convex one, we propose Lemma 1.

Lemma 1. *The equivalent of (14b) is expressed as*

$$1 - \frac{f_n^{loc} T_n^{max}}{L_n C_n} \leq x_n \leq \frac{T_n^{max} R_n}{L_n + R_n T_n^{proc}} \quad (15)$$

Proof: The proof of Lemma 1 is provided in Appendix A. ■

Assuming fixed $\mathbf{p}, \mathbf{w}, \mathbf{a}, \mathbf{f}$, $\mathcal{P}0$ can be decoupled into N sub-problems that can be independently optimized. By leveraging Lemma 1, the following optimization problem is formulated for each task:

$$\mathcal{P}1: \min_{x_n} E_n(x_n) \quad (16a)$$

$$\text{subject to } 1 - \frac{f_n^{loc} T_n^{max}}{L_n C_n} \leq x_n \leq \frac{T_n^{max} R_n}{L_n + R_n T_n^{proc}} \quad (16b)$$

The first derivative of $\mathcal{P}1$'s objective function is

$$\frac{\partial E_n(x_n)}{\partial x_n} = E_n^{off} - E_n^{loc} \quad (17)$$

According to (17), the objective function is monotonically increasing or decreasing based on the sign of the first derivative. By exploiting this monotonicity, x_n can be set to the lower/upper bound of (16b) when the objective function is increasing/decreasing. Therefore, the following Theorem is proposed:

Theorem 1. *The optimal offloading decision is obtained as*

$$x_n^* = \begin{cases} \max\{0, 1 - \frac{f_n^{loc} T_n^{max}}{L_n C_n}\}, & \text{if } E_n^{off} - E_n^{loc} \geq 0, \\ \min\{1, \frac{T_n^{max} R_n}{L_n + R_n T_n^{proc}}\}, & \text{otherwise} \end{cases} \quad (18)$$

Algorithm 1 Bisection Method for Finding p_n

Input: Maximum transmission power P_n^{max}
Output: Optimal p_n

- 1: Set $p_n^{LB} = 0$ and $p_n^{UB} = P_n^{max}$
- 2: **repeat**
- 3: Set $X = \frac{p_n^{LB} + p_n^{UB}}{2}$
- 4: **if** $\frac{\partial L_{p,w}(p_n^{LB})}{\partial p_n} \cdot \frac{\partial L_{p,w}(X)}{\partial p_n} < 0$ **then**
- 5: $p_n^{UB} = X$
- 6: **else**
- 7: $p_n^{LB} = X$
- 8: **end if**
- 9: **until** $|p_n^{UB} - p_n^{LB}| < 0.001$
- 10: **Return** p_n

B. Optimizing Wireless Resources while Fixing the Rest Optimization Variables

After obtaining the optimal offloading decision for each task, we consider $\mathbf{x}, \mathbf{a}, \mathbf{f}$ to be fixed in order to determine the optimal power and bandwidth allocation. Consequently, $\mathcal{P}2$ is formulated as

$$\mathcal{P}2 : \min_{\mathbf{p}, \mathbf{w}} \sum_{n=1}^N E_n(p_n, w_n) \quad (19a)$$

subject to:

$$T_n(p_n, w_n) \leq T_n^{max}, \forall n \quad (19b)$$

$$0 \leq p_n \leq P_n^{max}, \forall n \quad (19c)$$

$$0 \leq w_n \leq 1, \forall n \quad (19d)$$

$$\sum_{n=1}^N w_n \leq 1 \quad (19e)$$

To find the optimal power and bandwidth allocation, we employ the Lagrange multiplier and subgradient methods. Consequently, the respective Lagrangian of $\mathcal{P}2$ is obtained by (22) shown at the top of the next page. In (22), set $\mathcal{X}_{p,w} = \{\beta_n, \lambda_n, \mu, \pi_n\}$ denotes the non-negative Lagrange multipliers. Therefore, the dual function is written as follows:

$$D_1(\mathcal{X}_{p,w}) = \min_{p_n, w_n} L(p_n, w_n, \mathcal{X}_{p,w}) \quad (20)$$

subject to: (19b) – (19e)

Consequently, the dual problem is expressed as

$$\max_{\beta_n, \lambda_n, \mu, \pi_n} D_1(\mathcal{X}_{p,w}) \quad (21)$$

subject to: $\mathcal{X}_{p,w} \succeq 0, \forall n$

In accordance to the Karush–Kuhn–Tucker (KKT) conditions, the derivative of the Lagrangian function with respect to p_n is provided in (23). Since it is challenging to obtain a closed-form expression for (23), we utilize the bisection method for finding the root. The bisection method for finding p_n is presented in Algorithm 1.

To obtain the optimal bandwidth allocation, we calculate the first derivative of the Lagrangian with respect to w_n . The

Algorithm 2 Subgradient Method for Optimizing \mathbf{p}, \mathbf{w}

Input: Maximum transmission power $P_n^{max}, \forall n$ and system bandwidth W
Output: Optimal \mathbf{p}, \mathbf{w}

- 1: Initialize $p_n = P_n^{max}$ and $w_n = \frac{W}{|N|}, \forall n$
- 2: Initialize the Lagrange multipliers: $\beta_n, \lambda_n, \mu, \pi_n$
- 3: set $t = 0$
- 4: **repeat**
- 5: **for** $n = 1$ **to** N **do**
- 6: Calculate p_n using Algorithm 1
- 7: Calculate w_n according to (25)
- 8: Update the Lagrange multipliers using (26) - (29)
- 9: **end for**
- 10: Set $\mathcal{E}[t] = \sum_{n=1}^N E_n(p_n, w_n)$
- 11: **until** $|\mathcal{E}[t] - \mathcal{E}[t-1]| < 0.01$
- 12: **Return** \mathbf{p}, \mathbf{w}

result is provided by (24), shown at the top of the next page. Solving for w_n , the root can be obtained by

$$w_n = \sqrt{\frac{x_n L_n(p_n + \lambda_n) \ln 2}{(\beta_n + \mu) W \log_2[1 + \frac{p_n d_{n,s}^{-\delta}}{\sigma^2}]}} \quad (25)$$

After obtaining the solution for problem D_1 through Algorithm 1 and (25), the Lagrange multipliers are updated as

$$\lambda_n^{t+1} = \left[\lambda_n^t + s_1 \left(\frac{x_n L_n}{R_n} + x_n T_n^{proc} - T_n^{max} \right) \right]^+ \quad (26)$$

$$\pi_n^{t+1} = \left[\pi_n^t + s_2 (p_n - P_n^{max}) \right]^+ \quad (27)$$

$$\beta_n^{t+1} = \left[\beta_n^t + s_3 (w_n - 1) \right]^+ \quad (28)$$

$$\mu^{t+1} = \left[\mu^t + s_4 \left(\sum_{n=1}^N w_n - 1 \right) \right]^+ \quad (29)$$

where s_1, s_2, s_3 , and s_4 are the positive step sizes. The subgradient method for optimizing the wireless resource allocation is presented in Algorithm 2.

C. Optimizing Computing Resources while Fixing the Rest Optimization Variables

Having obtained the optimal offloading decision and wireless resource allocation, we will determine the optimal MEC assignment, as well as the optimal MEC frequency allocation to each task. Therefore, in this step, $\mathbf{x}, \mathbf{p}, \mathbf{w}$ are assumed to be fixed. Also, to address the non-convexity introduced by the binary constraint (14g), we relax it by setting it in $[0, 1]$ range. This relaxation can be perceived as dividing the offloaded portion of the task into multiple parts and processing them

$$L_{p,w}(p_n, w_n, \mathcal{X}_{p,w}) = \sum_{n=1}^N E_n(p_n, w_n) + \sum_{n=1}^N \lambda_n T_n(p_n, w_n) + \sum_{n=1}^N \pi_n (p_n - p^{max}) + \sum_{n=1}^N \beta_n (w_n - 1) + \mu \left(\sum_{n=1}^N w_n - 1 \right) \quad (22)$$

$$\frac{\partial L_{p,w}(p_n, w_n, \mathcal{X}_{p,w})}{\partial p_n} = \pi_n + \frac{x_n L_n (\log_2 [1 + \frac{p_n d_{n,s}^{-\alpha}}{\sigma^2}] - \frac{p_n + \lambda_n}{\ln 2 (p_n + d_{n,s}^{-\alpha} \sigma^2)})}{w_n W \log_2 [1 + \frac{p_n d_{n,s}^{-\alpha}}{\sigma^2}]^2} \quad (23)$$

$$\frac{\partial L_{p,w}(p_n, w_n, \mathcal{X}_{p,w})}{w_n} = \beta_n + \mu - \frac{x_n L_n (p_n + \lambda_n)}{w_n R_n} \quad (24)$$

in different RSUs. Consequently, $\mathcal{P}3$ is expressed as

$$\mathcal{P}3 : \min_{\mathbf{a}, \mathbf{f}} \sum_{n=1}^N E_n(a_{n,s}, f_{n,s}) \quad (30a)$$

subject to:

$$T_n(a_{n,s}, f_{n,s}) \leq T_n^{max}, \forall n \quad (30b)$$

$$0 \leq a_{n,s} \leq 1, \forall n, s \quad (30c)$$

$$\sum_{n=1}^N a_{n,s} \leq 2, \forall s \quad (30d)$$

$$0 \leq f_{n,s} \leq 1, \forall n, s \quad (30e)$$

$$\sum_{n=1}^N f_{n,s} \leq 1, \forall s \quad (30f)$$

To solve $\mathcal{P}3$, the Lagrange multiplier and subgradient methods can be employed. The Lagrangian of $\mathcal{P}3$ is given by (39), shown at the top of the next page. In (39), set $\mathcal{X}_{a,f} = \{\kappa_n, \lambda_{n,s}, \mu_s, \xi_{n,s}, \tau_s\}$ denotes the non-negative Lagrange multipliers. Thus, the dual function is written as follows:

$$D_2(\mathcal{X}_{a,f}) = \min_{a_{n,s}, f_{n,s}} L_{a,f}(a_{n,s}, f_{n,s}, \mathcal{X}_{a,f}) \quad (31)$$

subject to: (30b) – (30f)

Consequently, the dual problem is expressed as

$$\max_{\kappa_n, \lambda_{n,s}, \mu_s, \xi_{n,s}, \tau_s} D_2(\mathcal{X}_{a,f}) \quad (32)$$

subject to: $\mathcal{X}_{a,f} \succeq 0, \forall n, s$

To obtain the optimal task assignment, we take the first derivative of $L_{a,f}(a_{n,s}, f_{n,s}, \kappa_n, \lambda_{n,s}, \mu_s)$ with respect to $a_{n,s}$. According to (40), the n -th task can be assigned to the s -th RSU as follows:

$$a_{n,s} = 1 | s = \arg \min_s \frac{\partial L_{a,f}(a_{n,s}, f_{n,s}, \mathcal{X}_{a,f})}{\partial a_{n,s}} \quad (33)$$

Using (33), binary values for $a_{n,s}$ can be obtained without introducing errors due to the relaxation of (14g).

On the other hand, we utilize the bisection method shown in Algorithm 3 to obtain the optimal frequency scaling.

Algorithm 3 Bisection Method for Finding $f_{n,s}$

Input: Maximum RSU frequency F_s^{max}

Output: Optimal $f_{n,s}$

- 1: Set $f_{n,s}^{LB} = 0$ and $f_{n,s}^{UB} = 1$
 - 2: **repeat**
 - 3: Set $X = \frac{f_{n,s}^{LB} + f_{n,s}^{UB}}{2}$
 - 4: **if** $\frac{\partial \mathcal{L}_{a,f}(w_n^{UB})}{\partial f_{n,s}} \cdot \frac{\partial \mathcal{L}_{a,f}(X)}{\partial f_{n,s}} < 0$ **then**
 - 5: $f_{n,s}^{UB} = X$
 - 6: **else**
 - 7: $f_{n,s}^{LB} = X$
 - 8: **end if**
 - 9: **until** $|f_{n,s}^{UB} - f_{n,s}^{LB}| < 0.001$
 - 10: **Return** $f_{n,s}$
-

After problem D_2 is solved and the optimal task assignment and frequency vectors are obtained, the Lagrange multipliers are updated as

$$\lambda_n^{t+1} = \left[\lambda_n^t + s_1 \left(\frac{x_n L_n}{R_n} + x_n T_n^{proc} - T_n^{max} \right) \right]^+ \quad (34)$$

$$\kappa_{n,s}^{t+1} = \left[\kappa_{n,s}^t + s_2 (a_{n,s} - 1) \right]^+ \quad (35)$$

$$\mu_s^{t+1} = \left[\mu_s^t + s_3 \left(\sum_{n=1}^N a_{n,s} - 2 \right) \right]^+ \quad (36)$$

$$\xi_{n,s}^{t+1} = \left[\xi_{n,s}^t + s_4 (f_{n,s} - 1) \right]^+ \quad (37)$$

$$\tau_s^{t+1} = \left[\tau_s^t + s_5 \left(\sum_{n=1}^N f_{n,s} - 1 \right) \right]^+ \quad (38)$$

where s_1, s_2, s_3, s_4 and s_5 are the positive step sizes. The subgradient method for optimizing the computing resource allocation is presented in Algorithm 4.

D. Iterative Optimization using Block Coordinate Descent

The solution to the joint optimization problem is achieved by iteratively optimizing the subproblems. The employed BCD method is presented in Algorithm 5. During the initialization phase, the initial values for the optimization variables and the

$$L_{a,f}(a_{n,s}, f_{n,s}, \mathcal{X}_{a,f}) = \sum_{n=1}^N E_n(a_{n,s}, p_{n,s}) + \sum_{n=1}^N \lambda_n T_n(a_{n,s}, p_{n,s}) + \sum_{n=1}^N \sum_{s=1}^S \kappa_{n,s} (a_{n,s} - 1) + \sum_{s=1}^S \mu_s (\sum_{n=1}^N a_{n,s} - 2) + \sum_{n=1}^N \sum_{s=1}^S \xi_{n,s} (f_{n,s} - 1) + \sum_{s=1}^S \tau_s (\sum_{n=1}^N f_{n,s} - 1) \quad (39)$$

$$\frac{\partial L_{a,f}(a_{n,s}, f_{n,s}, \mathcal{X}_{a,f})}{\partial a_{n,s}} = \kappa_{n,s} + \mu_s + \frac{x_n L_n C_n (\lambda_n + (f_{n,s} F_s^{max})^3 \phi_s)}{f_{n,s} F_s^{max}} \quad (40)$$

$$\frac{\partial L_{a,f}(a_{n,s}, f_{n,s}, \mathcal{X}_{a,f})}{\partial f_{n,s}} = \xi_{n,s} + \tau_s + \frac{a_{n,s} x_n C_n L_n (-\lambda_n + 2(f_{n,s} F_s^{max})^3 \phi_s)}{f_{n,s}^2 F_s^{max}} \quad (41)$$

Algorithm 4 Subgradient Method for Optimizing \mathbf{a}, \mathbf{f}
Input: Maximum RSU frequency $F_s^{max}, \forall s$
Output: Optimal \mathbf{a}, \mathbf{f}

- 1: Initialize $a_{n,s} = 1$ and $f_{n,s} = 1, \forall n, s$
- 2: Initialize the Lagrange multipliers: $\kappa_n, \lambda_{n,s}, \mu_s, \xi_{n,s}, \tau_s$
- 3: set $t = 0$
- 4: **repeat**
- 5: **for** $n = 1$ **to** N **do**
- 6: Calculate $a_{n,s}$ using (33)
- 7: Calculate $f_{n,s}$ using Algorithm 3
- 8: Update the Lagrange multipliers using (34) - (38)
- 9: **end for**
- 10: Set $\mathcal{E}[t] = \sum_{n=1}^N E_n(a_{n,s}, f_{n,s})$
- 11: **until** $|\mathcal{E}[t] - \mathcal{E}[t-1]| < 0.01$
- 12: Return \mathbf{a}, \mathbf{f}

Algorithm 5 Block Coordinate Descent for the Joint Optimization

- 1: Initialize the optimization variables and the Lagrange multipliers
- 2: Set $t=0$;
- 3: **repeat**
- 4: Find $x_n, \forall n$ using (18)
- 5: Find p_n and w_n using Algorithm 2
- 6: Find $a_{n,s}$ and $f_{n,s}$ using Algorithm 4
- 7: Set $\mathcal{E}[t] = \sum_{n=1}^N E_n(x_n, p_n, w_n, a_{n,s}, f_{n,s})$
- 8: Set $t = t + 1$
- 9: **until** $t > t_{max}$ or $\frac{|\mathcal{E}[t] - \mathcal{E}[t-1]|}{\mathcal{E}[t-1]} < 0.01$
- 10: Return $\mathbf{x}, \mathbf{p}, \mathbf{w}, \mathbf{a}, \mathbf{f}$

Lagrange multipliers are set. In each step, the corresponding optimal value for each optimization variable is calculated and the algorithm ends after t_{max} iterations or if the energy consumption improvement is lower than 1%.

IV. PERFORMANCE EVALUATION

To evaluate the performance of our proposed solution, we utilize system-level Monte Carlo simulations. Table II summarizes the simulation parameters. The numbers of vehicles is set to $\{5, 10, 15, 20, 25\}$, while the number of RSUs is set to $\{1, 5, 10, 15, 20\}$. The maximum available transmission

power of each vehicle is 36 dBm, while the available system bandwidth is 20 MHz. Additionally, the path loss coefficient is set to 2 and 4, while noise variance is set to 10^{-8} . Regarding the computation model, the task size is uniformly distributed in the range $[500, 3500]$ Kbits, while the required cycles to process 1 bit and maximum latency are respectively set to 297.6 cycles/bit and 0.5s-3.5s ([30], [37], [46]). The energy consumption coefficients for the vehicles and RSUs are set to 10^{-28} . Furthermore, the computing frequency of vehicles ranges from 500 MHz to 800 MHz, while the maximum computing frequency of RSUs is set to 10 GHz.

Three evaluation scenarios are designed in order to highlight the impact of the allocation of wireless and computing resources on the total energy consumption in addition to the optimization of the offloading decision. In more detail, Scenario 1 is focused on optimizing only the offloading decision is optimized, whereas Scenario 2 is focused on optimizing the offloading decision and the allocation of wireless resources. Finally, in Scenario 3, the allocation of both wireless and computing resources is optimized in addition to the offloading decision.

TABLE II: Simulation Parameters

Parameter	Value
Number of vehicles N	$\{5, 10, 15, 20, 25\}$
Number of SCs S	$\{1, 5, 10, 15, 20\}$
Maximum vehicle transmission power P_n^{max}	36 dBm
Pathloss coefficient δ	$\{2, 4\}$
Noise variance σ^2	10^{-8}
Total system bandwidth W	20 MHz
Task size L_n	$[500, 3500]$ Kbits
Required cycles to process 1 bit C_n	297.6 cycles/bit
Maximum tolerable latency T_n^{max}	0.5s - 3.5 s
Consumption coefficients $\phi_n^{loc}, \phi_s^{mec}$	10^{-28}
Computing capability f_n^{loc}	$[500, 800]$ MHz
Maximum computing capability F_s^{max}	10 GHz

Fig. 2 shows the total energy consumption as a function of the number of vehicles, for various numbers of RSUs. The task size is randomly selected in the range $[500, 3500]$ Kbits, while the maximum delay tolerance is randomly selected in the range $[0.5, 3]$ seconds. In particular, Fig. 2-(a) shows the total energy consumption when the path loss exponent is

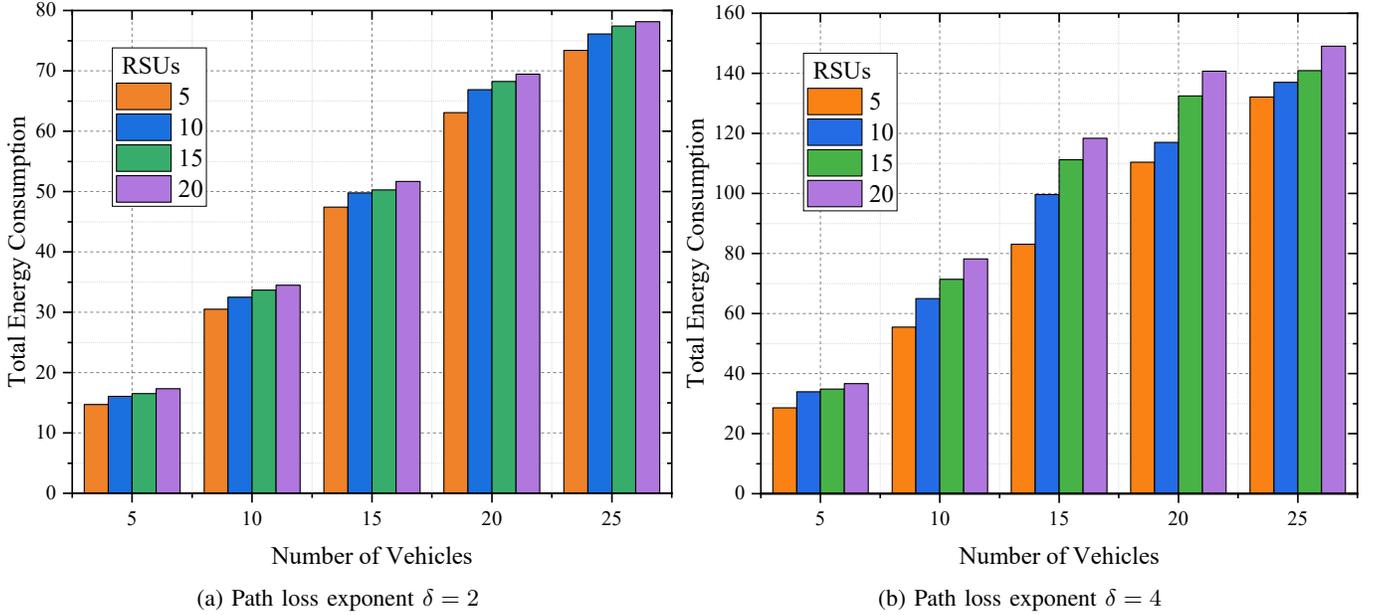


Fig. 2: Total energy consumption as a function of the number of vehicles for various numbers of RSUs

set to 2, whereas Fig. 2-(b) shows the corresponding energy consumption when the path loss exponent is set to 4. It is apparent that as the number of vehicles is increased, the total consumption also increases. This is expected because there exist more tasks to be computed leading to increased energy consumption. Furthermore, for a given number of vehicles, the total energy consumption is slightly increased as the number of RSUs increases. Thus, the number of RSUs has a small effect on the energy consumption for fixed vehicle numbers. Note that, the MEC-enabled RSUs have the capability to scale the allocated computing resources, therefore increasing the energy efficiency. With respect to the path loss exponent, for a given number of vehicles and RSUs, the total energy consumption of the system is increased as the path loss exponent increases. This is expected as additional power will be needed for uploading the respective tasks. Also, higher path loss will lead to lower channel capacity. Therefore, additional processing resources will be employed in order to timely process the task, resulting in higher energy consumption.

Fig. 3 presents the outage probability as a function of the number of vehicles, for various numbers of RSUs. The maximum latency is randomly selected in the range of $[0.5, 3.5]$ seconds, while the outage probability is calculated as the number of tasks that have not been computed in the required time to the total number of tasks. According to the results, the number of vehicles does not have a considerable impact on the outage probability. On the other hand, when there exist more RSUs, more tasks can be offloaded, leading to a reduced outage probability.

Fig. 4 shows the total energy consumption as a function of the maximum tolerable latency. The numbers of vehicles and RSUs are set to 20 and 10, respectively. Also, the energy consumption is evaluated for two cases of path loss exponents, particularly when $\delta = 2$ and $\delta = 4$. Based on the results, the total energy consumption is decreasing as the maximum

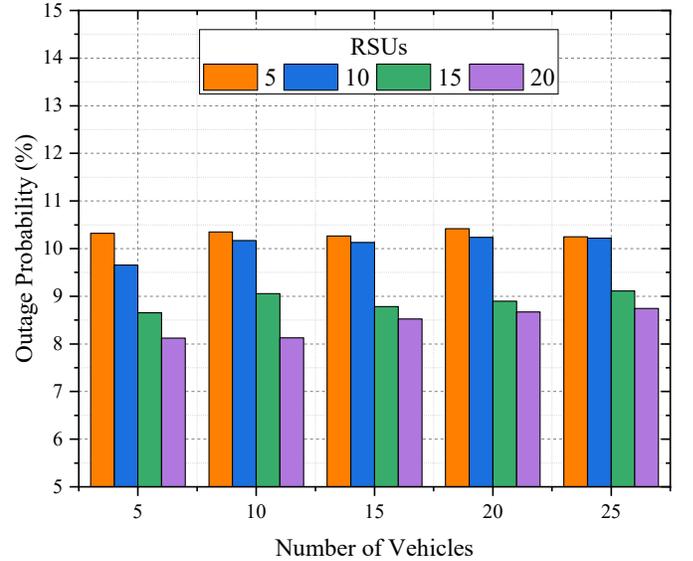


Fig. 3: Outage probability as a function of the number of vehicles for various numbers of RSUs

tolerable latency is increased. This is due to the fact that lower computing resources are allocated, leading to reduced energy consumption. As far as the task size is concerned, it is expected that when the task size is increased, more computing resources should be allocated, leading to increased energy consumption. Regarding the path loss exponents, the total energy consumption is increased for higher values of δ because of the additional transmission power and computing resources that will be employed.

Fig. 5 depicts a comparison between the three scenarios in terms of the total energy consumption for a varying number of devices. The task sizes are randomly selected in the range

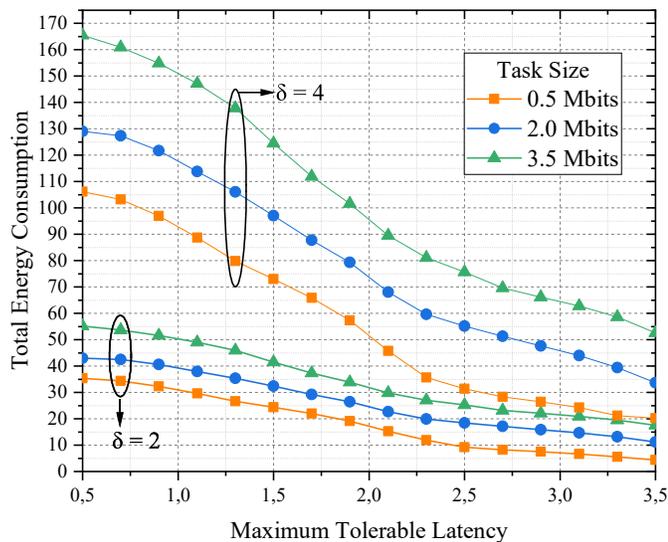


Fig. 4: Total energy consumption as a function of the maximum task latency for various task sizes

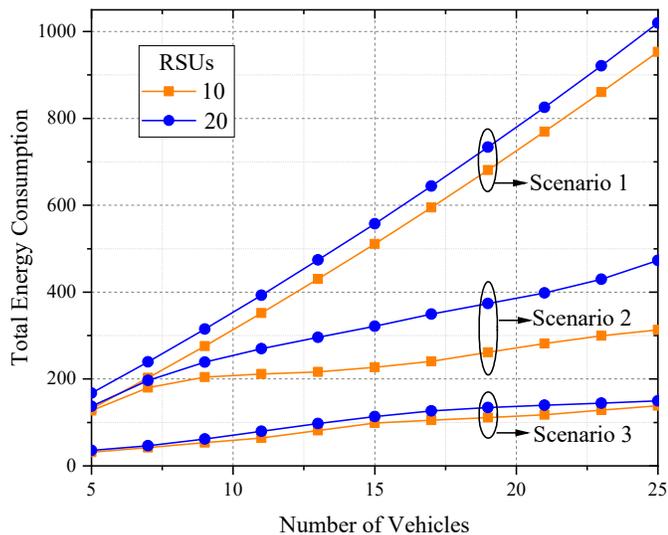


Fig. 5: Comparison between three scenarios in terms of the total energy consumption for varying number of devices

[500, 3500] Kbits, while the maximum delay tolerance values are randomly selected in the range [0.5, 3] seconds. Also, the number of RSUs is set to 10 and 20. In all cases, when the number of vehicles is increased the total energy consumption is also increased since there are more tasks to be processed, thereby consuming more energy (both for the upload and processing). Particularly, Scenario 1 results in the highest energy consumption as only the offloading decision is optimized. As a result, the vehicles transmit with the highest power (i.e., 36 dBm), while the RSUs process each task by assigning all available computing resources. On the other hand, Scenario 2 results in lower energy consumption as the allocation of wireless resources has been optimized, thus, lower levels of vehicle transmission power are required. Finally, Scenario 3 features the lowest total energy consumption since, in addition to the offloading decision, it optimizes the allocation of both

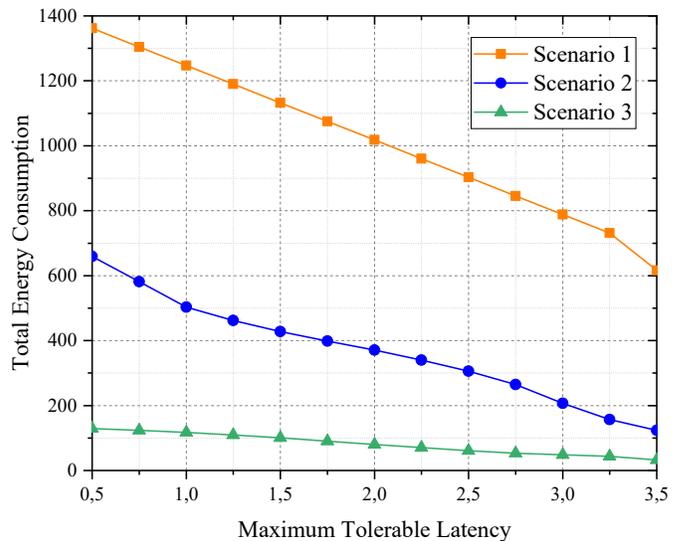


Fig. 6: Comparison between three scenarios in terms of the total energy consumption for varying tolerable delay

wireless and computing resources.

Finally, Fig. 6 shows a comparison between the scenarios with respect to the total energy consumption as a function of the maximum tolerable latency. The numbers of vehicles and RSUs are respectively set to 25 and 20, while the task sizes are randomly selected in the range [500, 3500] Kbits. Similarly to Fig. 4, the total energy consumption is decreased as the maximum tolerable latency of each task is increased. However, Scenario 1 features the highest overall energy consumption followed by Scenario 2, while Scenario 3 results in the lowest overall energy consumption. This is expected, as in Scenario 3 all the system variables are optimized, in contrast to Scenarios 2 and 3 where only a subset of the variables is optimized.

V. CONCLUSION

In this work, we considered the energy consumption minimization of a vehicular network. Specifically, we formulated the optimization problem as a joint optimization of the task offloading decision, the allocation of power and bandwidth, the assignment of tasks to MEC-enabled RSUs, and the frequency scaling of MEC-enabled RSUs. Since the optimization of the aforementioned problem is challenging, we decoupled it into three problems and leveraged the BCD method to iteratively optimize them. For the performance evaluation, we carried out system-level Monte Carlo simulations and evaluated the total energy consumption and the outage probability. The simulation results show that the proposed BCD-based approach can minimize the system energy consumption while maintaining a low outage probability. Moreover, three evaluation scenarios have been designed in order to highlight the impact of optimizing the allocation of both wireless and computing resources in addition to the offloading decision.

In the future, we aim to extend this work towards minimizing the average energy consumption over time, taking into account the mobility of the vehicles, as well as the arrival of new tasks. Furthermore, we aim to leverage our previous work

in [55] in order to incorporate UAVs to provide on-demand computation offloading. In this direction, the design of the offloading policy and resource allocation should also consider the limited energy reserves of the UAVs. Finally, in light of the exponential increase of internet of things devices, we will also evaluate the impact of novel multiple access methods in a scenario consisting of numerous devices and vehicles sharing the same wireless and computing resources.

APPENDIX A

For a given x_n the required time for the local computation is expressed as

$$x_n \left(\frac{L_n}{R_n} + T_n^{proc} \right) \leq T_n^{max} \Rightarrow x_n \leq \frac{T_n^{max} R_n}{L_n + R_n T_n^{proc}} \quad (42)$$

On the other hand, the required time for the offloaded computation is expressed as

$$(1 - x_n) \left(\frac{L_n C_n}{f_n^{loc}} \right) \leq T_n^{max} \Rightarrow x_n \geq 1 - \frac{f_n^{loc} T_n^{max}}{L_n C_n} \quad (43)$$

Combining (42) and (43) Lemma 1 is proved.

REFERENCES

- [1] I. Tomkos, D. Klouidis, E. Pikasis, and S. Theodoridis, "Toward the 6G network era: Opportunities and challenges," *IT Professional*, vol. 22, no. 1, pp. 34–38, Jan. 2020.
- [2] Y. Lu and X. Zheng, "6G: A survey on technologies, scenarios, challenges, and the related issues," *Journal of Industrial Information Integration*, vol. 19, p. 100158, Sep. 2020.
- [3] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, Mar. 2020.
- [4] J. Wang, K. Zhu, and E. Hossain, "Green internet of vehicles (IoV) in the 6G era: Toward sustainable vehicular communications and networking," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 391–423, 2022.
- [5] D. P. Moya Osorio, I. Ahmad, J. D. V. Sánchez, A. Gurtov, J. Scholliers, M. Kutila, and P. Porambage, "Towards 6G-enabled internet of vehicles: Security and privacy," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 82–105, 2022.
- [6] L.-M. Ang, K. P. Seng, G. K. Ijamaru, and A. M. Zungeru, "Deployment of IoV for smart cities: Applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2019.
- [7] A. Sacco, F. Esposito, and G. Marchetto, "Resource inference for sustainable and responsive task offloading in challenged edge networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1114–1127, Sep. 2021.
- [8] T. Lagkas, G. Papadimitriou, P. Nicopolitidis, and A. Pomportsis, "A new approach to the design of MAC protocols for wireless LANs: combining QoS guarantee with power saving," *IEEE Communications Letters*, vol. 10, no. 7, pp. 537–539, Jul. 2006.
- [9] L. D. Nguyen, H. D. Tuan, and T. Q. Duong, "Energy-efficient signalling in QoS constrained heterogeneous networks," *IEEE Access*, vol. 4, pp. 7958–7966, 2016.
- [10] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.
- [11] X. Xu, B. Shen, S. Ding, G. Srivastava, M. Bilal, M. R. Khosravi, V. G. Menon, M. A. Jan, and M. Wang, "Service offloading with deep q-network for digital twinning-empowered internet of vehicles in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1414–1423, 2022.
- [12] L. Zhao, K. Yang, Z. Tan, H. Song, A. Al-Dubai, A. Y. Zomaya, and X. Li, "Vehicular computation offloading for industrial mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7871–7881, Nov. 2021.
- [13] S.-S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020.
- [14] J. Luo, X. Deng, H. Zhang, and H. Qi, "Ultra-low latency service provision in edge computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [15] F. Brandherm, L. Wang, and M. Mühlhäuser, "A learning-based framework for optimizing service migration in mobile edge clouds," in *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking - EdgeSys '19*. ACM Press, 2019.
- [16] H. Wang, X. Li, H. Ji, and H. Zhang, "Federated offloading scheme to minimize latency in mec-enabled vehicular networks," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.
- [17] R. Yadav, W. Zhang, O. Kaiwartya, H. Song, and S. Yu, "Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 198–14 211, 2020.
- [18] G. Cui, Y. Long, L. Xu, and W. Wang, "Joint offloading and resource allocation for satellite assisted vehicle-to-vehicle communication," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3958–3969, 2021.
- [19] S. Chen, Y. Zheng, W. Lu, V. Varadarajan, and K. Wang, "Energy-optimal dynamic computation offloading for industrial IoT in fog computing," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 566–576, Jun. 2020.
- [20] H. Wang, J. Wang, G. Ding, L. Wang, T. A. Tsiftsis, and P. K. Sharma, "Resource allocation for energy harvesting-powered D2D communication underlying UAV-assisted networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 14–24, Mar. 2018.
- [21] Z. Ning, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3058–3067, May 2019.
- [22] W. Lu, P. Si, Y. Gao, H. Han, Z. Liu, Y. Wu, and Y. Gong, "Trajectory and resource optimization in OFDM-based UAV-powered IoT network," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1259–1270, Sep. 2021.
- [23] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint offloading and resource allocation in vehicular edge computing and networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.
- [24] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 158–11 168, Nov. 2019.
- [25] J. Sun, Q. Gu, T. Zheng, P. Dong, A. Valera, and Y. Qin, "Joint optimization of computation offloading and task scheduling in vehicular edge computing networks," *IEEE Access*, vol. 8, pp. 10466–10477, 2020.
- [26] 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.
- [27] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7635–7647, 2019.
- [28] X. Jianbin, H. Qingchun, A. Yaning, and W. Lu, "Joint task offloading and resource allocation in vehicle-assisted multi-access edge computing," *Computer Communications*, vol. 177, pp. 77–85, 2021.
- [29] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.
- [30] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.
- [31] Z. Ning, P. Dong, X. Wang, L. Guo, J. J. P. C. Rodrigues, X. Kong, J. Huang, and R. Y. K. Kwok, "Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1060–1072, 2019.
- [32] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26 652–26 664, 2019.
- [33] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus

- admm approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5087–5099, 2019.
- [34] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2019.
- [35] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3052–3056, 2019.
- [36] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [37] C. Chen, L. Chen, L. Liu, S. He, X. Yuan, D. Lan, and Z. Chen, "Delay-optimized V2V-based computation offloading in urban vehicular edge computing and networks," *IEEE Access*, vol. 8, pp. 18 863–18 873, 2020.
- [38] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 598–611, 2022.
- [39] W. Zhan, C. Luo, J. Wang, C. Wang, G. Min, H. Duan, and Q. Zhu, "Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5449–5465, 2020.
- [40] T. Lagkas, D. Klonidis, P. Sarigiannidis, and I. Tomkos, "Optimized joint allocation of radio, optical, and MEC resources for the 5G and beyond fronthaul," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4639–4653, 2021.
- [41] A. Ebrahimzadeh and M. Maier, "Cooperative computation offloading in FiWi enhanced 4G HetNets using self-organizing MEC," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4480–4493, Jul. 2020.
- [42] T. Sizer, D. Samardzija, H. Viswanathan, S. Thai Le, S. Bidkar, P. Dom, E. Harstead, and T. Pfeiffer, "Integrated solutions for deployment of 6G mobile networks," *Journal of Lightwave Technology*, vol. 40, no. 2, pp. 346–357, 2022.
- [43] H. Guo, Y. Wang, J. Liu, and N. Kato, "Super-broadband optical access networks (OANs) in 6G: vision, architecture, and key technologies," *IEEE Wireless Communications*, pp. 1–11, 2022.
- [44] C. Zhu, G. Pastor, Y. Xiao, and A. Ylajaaski, "Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 58–63, 2018.
- [45] H. Lu, Q. Liu, D. Tian, Y. Li, H. Kim, and S. Serikawa, "The cognitive internet of vehicles for autonomous driving," *IEEE Network*, vol. 33, no. 3, pp. 65–73, 2019.
- [46] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, "Selective offloading in mobile edge computing for the green internet of things," *IEEE Network*, vol. 32, no. 1, pp. 54–60, 2018.
- [47] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [48] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.
- [49] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016.
- [50] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.
- [51] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.
- [52] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.
- [53] D. Pliatsios, P. Sarigiannidis, S. Goudos, and G. K. Karagiannidis, "Realizing 5G vision through cloud RAN: technologies, challenges, and trends," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, May 2018.
- [54] S. Huang, C. Yang, S. Yin, Z. Zhang, and Y. Chu, "Latency-aware task peer offloading on overloaded server in multi-access edge computing system interconnected by metro optical networks," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5949–5961, Nov. 2020.
- [55] D. Pliatsios, P. Sarigiannidis, S. K. Goudos, and K. Psannis, "3D placement of drone-mounted remote radio head for minimum transmission power under connectivity constraints," *IEEE Access*, vol. 8, pp. 200 338–200 350, 2020.