Computer Networks 177 (2020) 107273

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

UAVs for traffic monitoring: A sequential game-based computation offloading/sharing approach

Ahmed Alioua[a,*], Houssem-eddine Djeghri[b], Mohammed Elyazid Tayeb Cherif[b], Sidi-Mohammed Senouci[c], Hichem Sedjelmaci[d]

[a] Department of Computer Science, University of Jijel - Mohamed Seddik Ben Yahia, Jijel, Algeria
[b] NTIC Faculty, University of Abdelhamid Mehri - Constantine 2, Constantine, Algeria
[c] DRIVE EA1859, University Bourgogne Franche-Comté, France
[d] Orange Labs, Châtillon, France

ARTICLE INFO

Keywords:
Unnamed arial vehicles
Vehicular networks
Computation offloading/sharing
Game theory

ABSTRACT

Recently, UAVs or Unnamed Aerial Vehicles have been proposed as flexible aerial support to assist ground vehicles for different applications such as rescue and traffic surveillance missions. UAVs can collect different data information about the road/traffic state usually as aerial photography and videos. The processing of this kind of data consists usually on pattern recognition and video processing which are complex tasks that necessitate powerful computing and energy resources. Unfortunately, the moderate UAV's computational and energy capabilities restrict local data processing. Fortunately, UAVs can leverage the computation resources of the surrounding edge network entities to enhance their computational capabilities. In this paper, we aim to achieve efficient data processing for the data collected by UAVs in the context of UAVs-aided vehicular networks for traffic monitoring missions. For this purpose, we propose a new system model where UAVs can offload and/or share intensive computation tasks with other nearby network nodes. Then, we use the computation response time, the energy consumed for the computation, the cost of cellular communication and the computation cost as the main system metrics to make any computation offloading/sharing decisions that optimize the system performance. We then modele the offloading/sharing decision-making problem as a sequential game, where we provide complete proof of the existence of the Nash equilibrium and propose an algorithm to reach such an equilibrium. The simulation results showed that the proposed game-based model outperforms other approaches by delivering better performance in terms of overall system utility with a data processing efficiency that varies between 43% and 97% depending on the computation approach, and provides a more efficient computation time and energy average.

1. Introduction

Vehicular networks are expected to play an important role in emerging smart cities for enhancing road safety and traffic efficiency.

Recently, the use of Unmanned Aerial Vehicles (UAVs) commonly known as drones have been democratized in diverse civilian applications and their number is significantly increasing in modern cities. This is principally propelled by their cheap price, easy deployment, low maintenance cost, and high-mobility [1]. In the last few years, UAVs are proposed as a suitable aerial support solution to facilitate many vehicular applications such as rescue missions and traffic surveillance. Thanks to the maturity of their underlying technology and with their three-dimensional-aerial mobility, UAVs offer additional degrees of freedom in aerial traffic surveillance, road safety improvement, and incident reporting compared to fixed roadside units (RSUs) [2,3]. Henceforth, UAVs can aerially cover large areas without any restriction related to the road ground network and congested traffic. For instance, in road traffic surveillance missions, UAVs are expected to detect, classify and identify objects or incidents/infractions on the road. UAVs collect diverse data information usually aerial photography such as image and video, compute and transmit the collected data to the nearby ground vehicular network in real-time, as it can store onboard these data for future use [4,5].

Generally, this kind of data in such an application needs to be processed rapidly to get relevant information about the traffic state and detect traffic infractions. For that, drones are often executing intensive computation tasks related to complex algorithms such as pattern recognition, video processing, and feature extraction. This type of

* Corresponding author.
E-mail addresses: ahmed.alioua@univ-jijel.dz (A. Alioua), houssem.djeghri@univ-constantine2.dz (H.-e. Djeghri), mohammed.tayebcherif@univ-constantine2.dz (M.E.T. Cherif), sidi-mohammed.senouci@u-bourgogne.fr (S.-M. Senouci), hichem.sedjelmaci@orange.com (H. Sedjelmaci).

https://doi.org/10.1016/j.comnet.2020.107273
Received 24 November 2019; Received in revised form 17 February 2020; Accepted 17 April 2020
Available online 19 May 2020

task requires heavy calculations, powerful computing and energy resources. Moreover, the size of the collected data can rapidly increase especially for a long sequence of high definition videos [5]. Unfortunately, the calculation of such heavy and intensive computation tasks by limited-resource devices such as UAVs can result in slow processing response time, long transmission delay and high energy consumption. Nonetheless, recently cloud-based solutions were adopted to address issues caused by the limited resources of UAVs [6,7]. This kind of solution, thanks to their powerful capabilities, can largely enhance the computation delay. However, they still suffering from high latency delay due to the far distance of their servers. Hence, fog computing with its edge servers is proposed as a more efficient computation solution [2,6]. Fortunately, UAVs can collaborate with theirs surrounding more powerful fog devices such as edge servers and ground vehicles by offloading heavy computation tasks in order to achieve more efficient data processing with an optimal balance between energy consumption and delay computation.

In the last few years, several studies have proposed solutions for the integration of UAVs to assist vehicular networks for different vehicular applications such as packet delivery [8-11], congestion detection [12] and road traffic monitoring [13]. However, none of these works have explored the manner how the UAV can collaborate with its surrounding edge node to optimize the processing of the collected data. In our previous works, we have addressed the problem of data processing in an infrastructure-less UAV-assisted software-defined vehicular network [5] and that of intensive-tasks computing in a UAV network [6].

In the same context, as an extension of our previous work [5], we address throughout this study the problem of how to achieve an efficient computation of intensive-computation tasks. These heavy tasks are related to the processing of the data collected by UAVs in the context of a UAV-aided road traffic surveillance scenario. We also formulate the data processing issue as a computation offloading/sharing decision-making problem using a game-theoretical based approach. Compared to [5], this work considers conjointly the computation response time, the UAV's energy consumed for the computation, the cost of using cellular communication and the remote computation cost as the main system metrics used to make any computation offloading/sharing decision, wherein [5] only computation delay and energy are considered in the offloading\sharing decision. It should be noticed that we are the first who proposed a system utility based on these innovative metrics. Also, we consider a multi-UAVs traffic monitoring scenario with more advanced computation strategies in the context of a vehicular system, counter to [5], that only considers a limited scenario with one UAV for a road accident rescue mission in the context of uncovered vehicular networks areas. The main contributions of this paper are summarized below.

- We propose a novel computation offloading/sharing policy for data processing in a UAV-aided road traffic monitoring scenario.
- We formulate the offloading/sharing computation decision-making problem as a three-player sequential game and design an algorithm to solve the problem.
- We conjointly consider a set of innovative system metrics: delay, energy, quality of the link, as well as communication and computation cost, to make any computation offloading\sharing decision. We evaluate the performances of the proposed game-based data processing policy for different system parameters.

The rest of the paper is organized as follows: we first briefly survey the related works on UAV-aided vehicular networks in Section 2. After, we present the problem formulation in Section 3 and we detail the computation model in Section 4. In Section 5, we detail the sequential computation-offloading/sharing game, prove the existence of Nash equilibrium and propose a distributed algorithm to reach this equilibrium. The simulation results are discussed in Section 6. Finally, Section 7 concludes the paper.

2. Related work

In the last few years, UAVs are proposed as a suitable aerial support solution to facilitate many vehicular applications such as search\rescue missions and road traffic monitoring. Almost all of the available works use small UAVs without embedded cameras to enhance packet delivery and data transmission in an uncovered vehicular environment without fixed infrastructure or with poor network coverage. For instance, Seliem et al. in [8] proposed a routing protocol that uses drones as a communication infrastructure to improve vehicular communications in order to achieve a minimum vehicle-to-UAV packet delivery delay. UAVs are used in [9-11] as an innovative routing solution to assist ground vehicles for efficiently enhance data delivery in vehicular urban areas. Shafiq et al. in [13] investigated a random-access transmission protocol for traffic monitoring applications in a high-way scenario, where a UAV is used to assist an RSU in order to enhance the system throughput. Authors in [14] proposed CTS-DP, an enhanced UAV-aided data dissemination protocol that employs drones as flying relays with data caching capability. CTS-DP aims to optimize trajectories of UAVs and achieve efficient coordination with RSUs, UAVs and ground vehicles for data dissemination. Oubbati et al. in [15] used a fleet of communicating UAVs to monitor traffic, detect, and localize road incidents. A backhaul based routing protocol that considers the high mobility and the restricted energy resources of UAV is proposed to help ground rescue teams to enhance the intervention time and efficiency. In [16], a drone solution combining RaptorQ-based content dissemination mechanisms is proposed in order to enhance content delivery in vehicular networks. Authors in [17] used UAVs to enhance the data communication performance of vehicular networks. For that, a UAV mobility model is presented and stochastic analysis of the end-to-end connectivity and data delivery delay is proposed in the presence of these UAVs. Sedjelmaci et al. in [18] proposed UVE, a new framework that used small UAVs as mobile infrastructure to enhance the connectivity between ground vehicles. A game-theoretical model is proposed to predict the road disconnection segments and mitigate the UAV energy of the draining effects. A novel simulation framework based on results retrieved from real testbed is proposed in [19] to measure the performance of UAV-to-ground vehicle communications.

Some other works use more advanced UAVs with embedded cameras to assist ground vehicles in traffic monitoring and road surveillance. Jian et al. [12] proposed to combine UAVs with artificial neural networks to detect traffic congestion and alert the authorities to improve road traffic. As well as [12], Khan et al. [20] proposed to use UAVs for road traffic monitoring, particularly for analyzing traffic flows at urban roundabouts. For this purpose, an analytical methodology has been presented for analyzing traffic flow conditions at signalized intersections. Zhu et al. [21] propose a drone-based method to detect and simulate pedestrian movements at intersections with frequent collisions. Using in-depth learning methods and high-resolution extraction of pedestrian, bicycle and vehicle movements, they proposed a model for saving the pedestrian lives based on theirs trajectories. Authors in [22] propose two algorithms to optimize the placement of UAVs charging docks, minimize the UAVs energy consumption and maximize the UAVs coverage area for a better road incident exploration. Kyrkou et al. [23] propose a solution based on artificial intelligence to obtain the best strategy that ensures the rapid deployment of UAVs in traffic monitoring missions. In [24], authors used a fleet of UAVs to monitor, collect and send road traffic information to a processing center for traffic regulation purposes. Unfortunately, all the above-cited works consider limited baseline data processing mechanisms for the data collected by UAVs, where they are either performed by the UAV or by a ground server. These works present a flagrant lack of intelligent mechanisms adapted for delay-sensitive scenarios, which can enhance the data processing processes in terms of computation delay and save UAV's processing energy.

Recently, Messous et al. in [2,6] and [25] have used game theoretical models to tackle the problem of offloading heavy computation tasks of UAVs while achieving the best possible tradeoff between energy consumption, computation delay and communication cost. In [26], the authors proposed a computation offloading framework for a UAV system in order to achieve an optimal tradeoff between computation delay and energy. In the proposed framework, intensive tasks are offloaded to edge servers based on computation load parameters and the state in real-time of the network's devices. Moreover, in our previous work [5], we have addressed the challenge of processing the collected data by UAV in the context of a road safety scenario for vehicular areas without fixed infrastructure coverage. We have formulated the problem as a decision-making problem in terms of computation offloading/sharing while trying to find the best tradeoff between computation delay and energy. A two-player sequential game based on SDN (Software-Defined Networking) paradigm was used to model the computation offloading/sharing problem. Nevertheless, in [5] we have proposed a partial solution with one UAV and few computation possibilities that only work in specific vehicular areas without fixed infrastructure coverage.

This work is an extension of our previous work in [5], where we use a fleet of UAVs to assist authority vehicles in the surveillance of road traffic. Unlike other related works, we focus on how to achieve an efficient data processing of the data information collected by UAVs. Moreover, we extend our game-theoretical model in [5] to use a multi-UAVs based computation offloading/sharing strategy with more advanced offloading/sharing possibilities based on a combination of new innovative system metrics (*i.e.*, the delay, the energy, the communication and computation cost).

To the best of our knowledge, we are the first to use a multi-players sequential game to investigate the problem of offloading/sharing computation related to data processing in the context of a multi UAVs-aided road traffic monitoring mission. Our game-theoretical model proposes a novel system utility-based not only on the commonly used system metrics such as the computation delay and energy but also on new innovative metrics such as computation cost and communication cost. As well as, we consider more advanced computation strategies to enhance the data processing process.

3. Problem formulation

Nowadays, with a considerable increase in the number of vehicles, road traffic monitoring and management is becoming a major challenge for the authority. To overcome this problem, some works [8-11,13-24] propose to equip authority vehicles, such as police and emergency vehicles, with UAVs to assist them in traffic monitoring operations. UAVs can detect vehicles that have committed traffic offenses such as speeding or crossing a continuous line [27]. In addition, UAVs can act as aerial road monitoring devices and help road surveillance authority to improve traffic fluidity by detecting and signaling congested road segments [24], as well as enhance road safety. The UAVs role consists of collecting different data information on the road state through their various sensors such as aerial images and videos. The collected data must be processed as soon as possible to extract relevant information on the road state. This information allows the road surveillance authority to make the appropriate decisions at the right time to preserve the safety and fluidity of road traffic. The processing of this kind of data is usually pattern recognition and video processing which well known that they are complex tasks with intensive computation tasks that necessitate powerful computing and energy resources.

In this section, we discuss the problem of offloading computation decision-making related to the UAV's data processing optimization in the context of UAV-aided road traffic monitoring missions. We use a fleet of UAVs to provide aerial assistance to ground road authority vehicles. Thus, we start by presenting the system model before describing the proposed data processing policy.

3.1. System model

The proposed system model is based on a scenario where a fleet of UAVs assists road authority vehicles during a traffic monitoring mission on a highway. Thanks to their flexibility, fluidity and embedded cameras, UAVs can be rapidly deployed at any point on the highway to detect vehicles that commit offenses. In addition to their ability to fly at altitude, UAVs are much less detectable than a static monitoring camera or speed radar. The information captured by the UAVs must be processed in a short time to allow the road surveillance authority to detect road offenses an then intervene quickly.

To model our approach, we consider an authority traffic monitoring vehicle (AV) equipped with a fleet of UAVs noted $D=\{1,2,\ldots,m\}$. In our model, both of vehicles and UAVs can communicate with an edge server using cellular link trough LTE-A/5G technology. UAVs are connected to each other's and communicate using the cellular vehicle to everything technology (C-V2X PC5, we simplify as C-V2X) [33] and with the road authority vehicle on the ground using a radio link. All vehicles use C-V2X to communicate via V2V (Vehicle to Vehicle) connection to each other's and use an LTE-A/5G interface to communicate with the infrastructure, see Fig. 1.

We assume that at any time at least the road authority vehicle is reachable by a UAV directly in one hop or in multi-hop mode, passing through other intermediate nodes. Since we focus our efforts on how to optimize the UAV's data processing, we neglect the UAV's energy of moving. Hence, we consider in our study only the computation and communication energy consumptions. We suppose also that every node detains and manages a table of its nearby nodes. This table contains the list of the nearby nodes and the quality of the link with these nodes. The nearby nodes table is periodically updated through exchanging beacon messages.

3.2. Data processing policy

The role of the UAVs, in addition to road traffic monitoring and data collection, is to detect possible traffic offenses or road accidents. UAVs collect diverse data information usually aerial photography such as image and video, process these data and transmit the resulted information to the road authority vehicles on the ground. Indeed, drones are often executing intensive computing tasks related to the processing of the collected data such as pattern recognition, video processing, and feature extraction. In addition, they are expected to rapidly process these collected data using the lowest possible energy consumption. This latter represents a very valuable resource for the success of the aerial monitoring operation. In fact, two types of UAVs are defined according to their role in the data processing operation: the primary UAV (PU) which is the node that collects the data and aims to optimize the corresponding computation tasks, and nearby secondary UAVs (NU) that can cooperate with the primary UAV to perform the computation.

To this purpose, we propose a varied set of computation strategies to efficiently process the UAVs data information. Effectively, the primary UAV can:

(1) Perform the computation tasks locally using its own resources.
(2) Share a part of the computation tasks with one or more of its nearby secondary UAVs. Drones can cooperate to reduce the computation cost in terms of delay and energy.
(3) Offload the computation tasks to an edge server to enhance it computational capability, since this latter has more powerful computational resources, in addition to the advantage of edge deployment which reduces the transmission delay.
(4) Offload the computation tasks to the road authority vehicle. Because the ground vehicles are supposed have higher computational capabilities than UAVs and can lead to better computation performance in some scenarios.

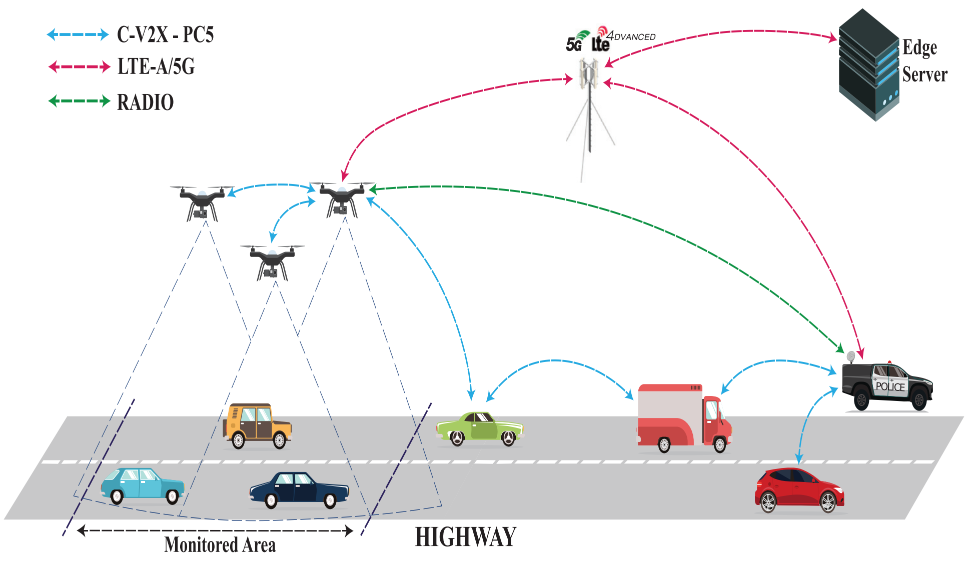


Fig. 1. UAV-aided HetVNet for a road traffic monitoring scenario.

The road authority vehicle has the following computational possibilities when it receives an offloaded task from a UAV:

(1) Perform the computation tasks locally.
(2) Share the computation tasks with other nearby vehicles (NV). Here, authority vehicle can cooperate with it nearby vehicles to enhance the task computation.
(3) Offload the computation tasks to an edge server. Because the computational capability of the edge server is more powerful than that of the authority vehicle.

The nearby vehicle has the following computational possibilities when it has to process a data computation task shared by the road authority vehicle:

(1) Accept to process the data shared by the road authority vehicle, if it is idle.
(2) Refuse to process the data shared by the authority vehicle, when its resources are busy by calculating its own tasks.

A possible illustration example of the effectiveness of the proposed computation strategies is given as follow. In some scenarios, to save the energy of the UAVs and decrease the computing response delay, it could be preferable to take advantage from the powerfull computation resources of the edge server and offloading high intensive tasks with small data (extracting some features from a picture) to be computed by the edge server and send back only the computation results with small size than computing this kind of tasks locally using the limited resource UAVs. However in some other scenarios, when for instance going to process and compute less intensive tasks with heavy data size (simple processing of a high-quality video), the local computation on the drone can result in better system performance in terms of computation delay and energy. Here offloading\sharing computation could result in high transmission energy and delay. Moreover, sharing computation between nearby UAVs could give good performances for tasks with moderate computational cycles and data size, taking benefit from the collaborative distributed computation and the near distance of drones in addition of the free of charge communication between UAVs.

In our data processing policy, the primary UAV initially collects different types of data information (usually aerial photography such as photos and videos) on the road segment that it is supposed to monitor. Then, it makes a computation offloading/sharing decision when it starts to process a data computational task i ($d_i^u = \{0, 1, 2, 3\}$): where, ($d_i^u = 0$) if it chooses to execute the computational task locally, ($d_i^u = 1$) if it chooses to offload the computational task to the edge server using LTE-A/5 G, ($d_i^u = 2$) if it chooses to share the computational task with other nearby UAVs using C-V2X interface, and ($d_i^u = 3$) if it chooses to offload the computational task to the road authority vehicle using the C-V2X interface. Once the road authority vehicle receives a computational task, it must decide whether to perform the calculation locally ($d_i^{av} = 0$), offloads the computation to an edge server ($d_i^{av} = 1$), or share the computation with a nearby vehicle ($d_i^{av} = 2$). The nearby vehicle when it receives a shared computational task from the road authority vehicle, it has to choose between two possibilities, either it accepts to compute the task ($d_i^{nv} = 0$) or deny the computation request ($d_i^{nv} = 1$). The data processing policy is illustrated in Fig. 2.

4. Computation model

For the computation model, we consider that the collected data can be divided into a set of intensive computation tasks. Similar to existing related works in [2,5,6,25,38], and [39], we assume that the computational tasks can be partitioned into inter-dependent sub-tasks with arbitrary granularity using dynamic partial partitioning technique [37]. We model the task i by (C_i, Ds_i, Rs_i, S_i, Rv_i), $i \in N$, this representation is slightly inspired of that proposed by Liwang et al. in [28]. Where C_i represents the total number of processor computation cycles required to complete task i, Ds_i represents the total size of the task's data, the data includes the input parameters necessary for the computation and the program code to be executed, Rs_i represents the size of the computation results to be sent back to the road authority vehicle, and Si represents the level of sensitivity to delay of task i, where $S_i \in \{0, 1\}$: 0 indicates that task i is not delay-sensitive, 1 indicates that task i is delay-sensitive. Rv_i represents the relevance degree of the obtained results of task i regarding the initial waited computation goal. Where $Rv_i \in \{0, 1\}$, 0 indicates that the obtained results from the computation of task i are not relevant and 1 indicates that results are relevant. Initially, we suppose that the analysis results are relevant ($Rv_i = 1$) and after the end of the computation and the reception of the analysis results by the authority vehicle, a road traffic monitoring agent, can decide if the results are relevant or not by modifying the value of Rv_i.

Without loss of generality and similar to many previous similar studies dealing with computation offloading problem in mobile cloud/edge computing and in UAV networks (see [2,5,6,25,38], and [39], we make the common assumption that the network runs based on a time-slotted manner to enable tractable analysis. Where the time is partitioned into equal small intervals or slots. Each interval represents one decision slot within it only one task is processed and all network parameters (*e.g.*, the number and position of UAVs\vehicles, as well as the quality of wireless channels) remain unchanged, during a decision time slot, while they

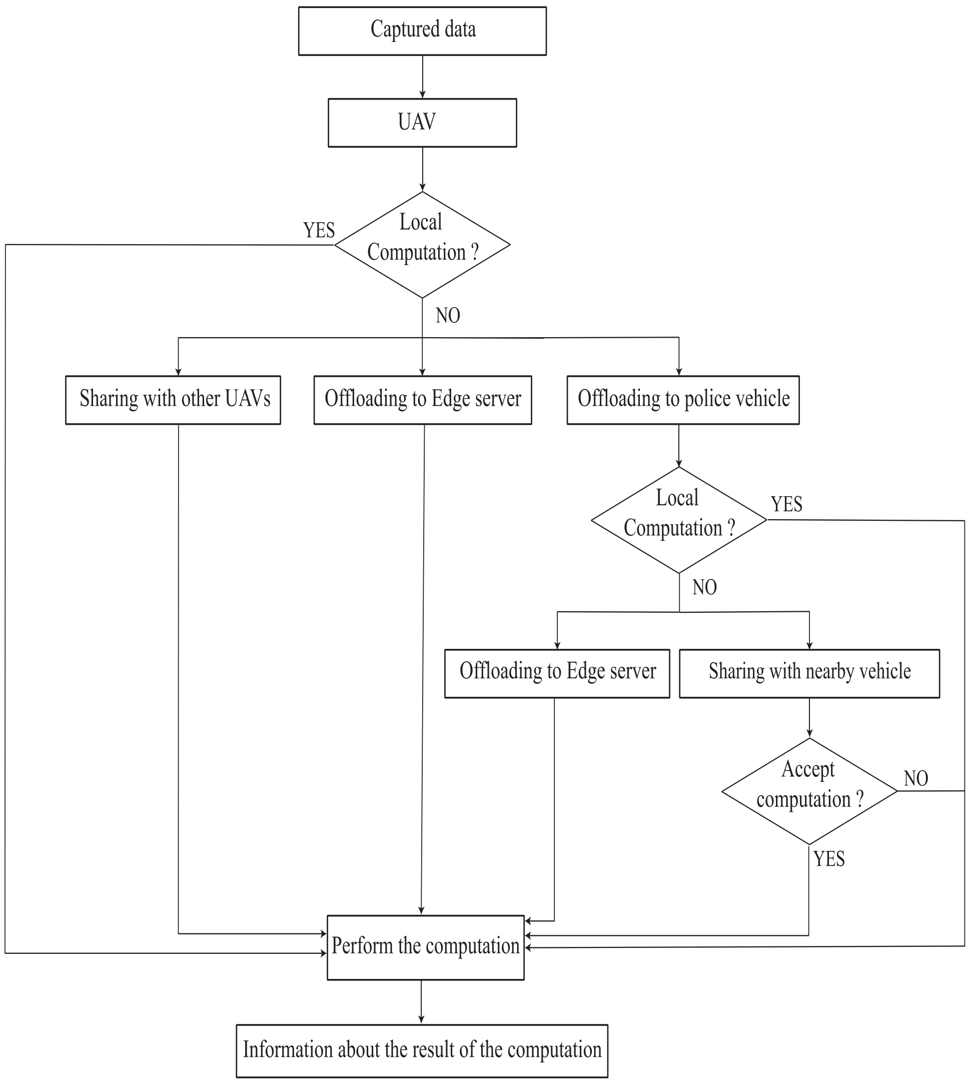

Captured data
UAV
Local Computation ?
YES
NO
Sharing with other UAVs
Offloading to Edge server
Offloading to police vehicle
Local Computation ?
YES
NO
Offloading to Edge server
Sharing with nearby vehicle
Accept computation ?
NO
YES
Perform the computation
Information about the result of the computation

Fig. 2. Data processing policy flowchart.

may move throughout different periods [40]. For this reason, we omit the UAV hovering energy in our computational model.

During the road monitoring mission, the collected data must be processed quickly in order to allow the road authority vehicle to intervene at the right time. Furthermore, optimizing UAVs energy is essential for the success of the monitoring mission. In addition, sending data via the LTE-A/5 G interface requires to pay a cost for using cellular technology, which imposes a rationalization of communication cost. Similarly, requesting an edge server or a ground vehicle for computation services requires additional service charges. As a result, the computation response time, the energy consumed for the computation, the cost of cellular communication and the computation cost are the main metrics that we considered in our computation approach to make any computation offloading/sharing decision. It should be noticed that we are the first who proposed a system utility based on these innovative metrics.

In the following, we will detail the computation model of our data processing system based on similar models proposed in [2,5,6,25,28], and [34].

4.1. Primary UAV's computation model

Based on the delay and energy required to compute the tasks, as well as the communication cost and remote computing cost, the UAV aims to make the best decision when it has to process the collected data. Indeed, the UAV can perform the computation locally, share the computation with nearby UAVs or offload the computation to a more powerful node, either to the road authority vehicle or even to a computation edge server.

(1) Local computation

The total execution time ($T_{i,l}^{pu}$) of local computation of task i by the primary UAV, is equal to the sum of the local execution time by the primary UAV (T_i^{l-pu}) plus the time for sending the results to the road authority vehicle ($T_{i,l-rs}^{pu}$). The computation results are sent to the road authority vehicle either in one-hop if the road authority vehicle is within the coverage area of the primary controller or in multi-hop via other intermediate vehicles using the C-V2X interface [5]. $T_{i,l}^{pu}$ is given by:

$$T_{i,l}^{pu} = T_{i,l}^{l-pu} + T_{i,l-rs}^{pu} = \left(\frac{C_i}{F_{CPU}^{Pu}}\right) + \left(\sum_{k=1}^{h} \frac{Rs_i}{R_{C-V2X}}\right) \quad (1)$$

Here F_{CPU}^{pu} represents the computation frequency of the primary UAV in processor cycles per second and R_{C-V2X} represents the data transmission rate through the C-V2X interface. h is the number of V2V hops required to reach the road authority vehicle.

The total energy of the local computation by the primary UAV ($E_{i,l}^{pu}$) is equal to the sum of the local computation energy (E_i^{l-pu}) plus the energy it consumes to send back the results to road authority vehicle ($E_{i,l-rs}^{pu}$). We assume that vehicles do not have energy constraints; therefore, we neglect the energy consumption of vehicles to perform the computation task and send back the results. $E_{i,l}^{pu}$ is given by:

$$E_{i,l}^{pu} = E_i^{l-pu} + E_{i,l-rs}^{pu} = \left(C_i \times e_{CPU}^{pu}\right) + \left(R_{Si} \times e_{\mathrm{C-V2X}}^{pu}\right) \tag{2}$$

Here, e_{CPU}^{pu} represents the energy consumed per processor computation cycle of the primary UAV, and e_{C-V2X}^{pu} represents the energy consumed by the C-V2X interface to send back one data unit to the road authority vehicle.

(2) Offloading computation to the edge server

The total time of offloading computation to the edge server ($T_{i,o-es}^{pu}$) is equal to the sum of the time required to send the data task i (Ds_{i}) to the edge server ($T_{i,t-es}^{pu}$) plus the local computation time on the edge server (T_i^{l-es}) plus the time for sending the results to the road authority vehicle ($T_{i,o-rs}^{pu}$). $T_{i,o-es}^{pu}$ is given by:

$$\begin{aligned} T_{i,o-es}^{pu} = & \quad T_{i,t-es}^{pu} + T_i^{l-es} + T_{i,o-rs}^{pu} \\ = & \left(\frac{Ds_i}{R_{\mathrm{LTE-A/5G}}^{pu}}\right) + \left(\frac{C_i}{F_{CPU}^{es}}\right) + \left(\frac{Rs_i}{R_{\mathrm{LTE-A/5G}}^{es}} + \sum_{k=1}^{h} \frac{Rs_i}{R_{\mathrm{LTE-A/5G}}^{es}}\right) \end{aligned} \tag{3}$$

Here F_{CPU}^{es} represents the computation frequency of the edge server in processor cycles per second, ($R_{LTE-A/5G}^{pu}$) and ($R_{LTE-A/5G}^{es}$) represent the data transmission rate through the LTE-A/5 G interface of the primary UAV and edge server, respectively.

The total energy consumption of the offloading computation by the edge server ($E_{i,o-es}^{pu}$) is equal to the energy for sending the data of task *i* (Ds_i) to the edge server ($E_{i,t-es}^{pu}$) plus the energy for sending the results to the road authority vehicle ($E_{i,o-rs}^{pu}$). We assume that the edge server has no energy constraint; therefore, we neglect the energy consumption of the edge server to perform the computation and send back the results to the UAV. $E_{i,o-es}^{pu}$ is given by:

$$E_{i,o-es}^{pu} = E_{i,t-es}^{pu} + E_i^{Rs} = \left(Ds_i \times e_{LTE-A/5G}^{pu}\right) + \left(Rs_i \times e_{\mathrm{C-V2X}}^{pu}\right) \tag{4}$$

Here, $e_{LTE-A/5G}^{Pu}$ represents the energy consumed by the LTE-A/5 G interface of the UAV to send one data unit to the road authority vehicle.

Using cellular communication through the LTE-A/5 G interface requires to pay a communication cost depending on the size of the data sent, unlike communications using the C-V2X interface which are free [29]. Therefore, we calculate here the cost of sending the data of task *i* via LTE-A/5 G to the edge server ($\Pi_{i,pu}^{cell,\ o-es}$). This cost is generally a monetary charge. Similarly, we assume that the edge server uses LTE-A/5 G technology for free to send the data. ($\Pi_{i,pu}^{cell,o-es}$) is given by:

$$\Pi_{i,pu}^{cell,o-es} = Ds_i \times \pi_{LTE-A}^{cell} \tag{5}$$

Here, π_{LTE-A}^{cell} refers to the communication price of sending one data unit via the LTE-A/5 G interface.

The computation of a task offloaded by the primary UAV to the edge server also inflicts a service computation cost related to the complexity of the task to be computed. Therefore, we calculate the cost of computing a task *i* by the edge server ($\Pi_{i,pu}^{cpt,o-es}$) using the following equation:

$$\Pi_{i,pu}^{cpt,o-es} = C_i \times \pi_{es}^{cpt} \tag{6}$$

Here, π_{es}^{cpt} refers to the computation price of one processor cycle of an offloaded task by the edge server.

(3) Sharing computation with nearby UAVs

The primary UAV can share the computation of data tasks with its nearby UAVs in order to better optimize the system's metrics. To do this, the primary UAV first negotiates with nearby UAVs on the percentage of task computation cycles that it can share the calculation. As a result, the total time required for shared computation with nearby UAVs ($T_{i,s-nu}^{pu}$) is equal to the sum of the negotiation time with the nearby UAVs on the portion of computational cycles to be shared (T_i^{ng}) plus the transmission time of the partial task's data to be shared with the nearby UAVs ($T_{i,t-nu}^{pu}$), plus the time of the local computation of the primary UAV (T_i^{l-pu}), plus the average local computation time of nearby UAVs (T_i^{a-nu}) plus the time needed to send back the results to the primary UAV by the nearby UAVs and then to the road authority vehicle ($T_{i,\ s-rs}^{pu}$). For a fleet of UAVs composed of *n* UAVs, we note the ratio $\sum_{j=1}^{n-1} \gamma_j$ which represents the sum of the computation ratios to be shared with the nearby UAVs, where *n-1* represents the number of nearby UAVs, and γ_{j} represents the computation ratio supported by a nearby UAV *j*, such that $0 < \sum_{j=1}^{n-1} \gamma_j < 1$ $T_{i,s-su}^{pu}$ is given by:

$$\begin{aligned} T_{i,s-su}^{pu} = & \quad T_i^{ng} + T_{i,t-nu}^{pu} + T_i^{l-pu} + T_i^{a-nu} + T_{i,s-rs}^{pu} \\ = & \left(\sum_{j=1}^{n-1} \gamma_j \times \frac{Ds_i}{R_{C-V2X}}\right) + \left(\left(1 - \sum_{j=1}^{n-1} \gamma_j\right) \times \frac{C_i}{F_{CPU}^{pu}}\right) + \left(\frac{T_i^{l-nu}}{n-1}\right) \\ & + \left(\sum_{j=1}^{n-1} \gamma_j \times \frac{Rs_i}{R_{C-V2X}} + \sum_{k=1}^{h} \frac{Rs_i}{R_{C-V2X}}\right) \end{aligned} \tag{7}$$

Where, $F_{CPU}^{j,nu}$ represents the computation frequency of a nearby UAV *j* in processor cycles per second. We suppose that UAVs can have different computational capabilities.

Total energy consumption of sharing computation with the secondary UAV ($E_{i,s-nu}^{pu}$) is equal to the sum of the energy consumed for local computation of the portion of task *i* by the primary UAV ($E_i^{(l-pu)}$) plus the energy of sending the shared part of the task data to the nearby UAVs ($E_{i,\ t-nu}^{pu}$), plus the energy of the local computation on nearby UAVs (E_i^{l-nu}), plus the energy consumed to send back the results to the primary UAV by the nearby UAVs and then to the road authority vehicle ($E_{i,s-rs}^{pu}$). $E_{i,s-nu}^{pu}$ is given by:

$$\begin{aligned} E_{i,s-nu}^{pu} = & \quad E_i^{l-pu} + E_{i,t-nu}^{pu} + E_i^{l-nu} + E_{i,s-rs}^{pu} \\ = & \left(\left(1 - \sum_{j=1}^{n-1} \gamma_j\right) \times C_i \times e_{CPU}^{pu}\right) + \left(\sum_{j=1}^{n-1} \gamma_j \times Ds_i \times e_{C-V2X}^{pu}\right) \\ & + \left(\sum_{j=1}^{n-1} \left(\gamma_j \times e_{CPU}^{j,nu}\right) \times C_i\right) + \left(\left(\sum_{j=1}^{n-1} e_{\mathrm{C-V2X}}^{j,nu} \times Rs_i\right) + Rs_i \times e_{\mathrm{C-V2X}}^{pu}\right) \end{aligned} \tag{8}$$

Here, $e_{CPU}^{j,nu}$ represents the energy consumed by one processor computation cycle of a nearby UAV *j*, and $e_{PC5}^{j,nu}$ represents the energy consumed by the C-V2X interface to send one data unit from a nearby UAV *j* to the primary UAV.

(4) Offloading computation to the road authority vehicle

The time required to offload a data task to be computed by the road authority vehicle ($T_{i,o-av}^{pu}$) is the time required to send data to the road authority vehicle which is defined by:

$$T_{i,o-av}^{pu} = \sum_{k=1}^{h} \frac{Ds_i}{R_{\mathrm{C-V2X}}} \tag{9}$$

The total energy consumed to offload the task's data to be computed by the road authority vehicle ($E_{i,o-av}^{pu}$) is represented by the energy consumed by the UAV to send the task's data *i* (Ds_i) to the road authority vehicle. $E_{i,o-av}^{pu}$ is given by:

$$E_{i,o-av}^{pu} = Ds_i \times e_{\mathrm{C-V2X}}^{pu} \tag{10}$$

4.2. Road authority vehicle computation model

When the road authority vehicle receives a computation task *i* offloaded by the primary UAV it chooses: either to compute the task locally, offload it to an edge server using the LTE-A/5 G interface, or share the computation with one of its nearby vehicles using the C-V2X interface.

(1) Local computation

The total time for the local computation ($T_{i,l}^{av}$) of task *i* is represented by the computation time of task *i* by the road authority vehicle which is defined by:

$$T_{i,l}^{av} = \frac{C_i}{F_{CPU}^{av}} \tag{11}$$

Where F_{CPU}^{av} represents the computation frequency of the road authority vehicle in processor cycles per second.

In our system, we assume that some vehicles on the road may have an electric or hybrid engine (a gasoline engine combined with an electric engine). The latter supplies their electric engines with on-board batteries, so energy optimization in electric and hybrid vehicles is becoming an important issue to maximize their autonomy on the road. For this purpose, we consider the computation energy consumption for electric or hybrid nearby vehicle and we neglect it for a gasoline vehicle. We note ϑ the type of vehicle, such as ϑ=0 if the vehicle is gasoline, and $\vartheta = 1$ if the vehicle is electric or hybrid. The energy consumed by the road authority vehicle during the local computation ($E_{i,l}^{av}$) is given by:

$$E_{i,l}^{av} = \vartheta \times \left(C_i \times e_{CPU}^{av}\right) \tag{12}$$

Where e_{CPU}^{av} represents the energy consumed by the road authority vehicle for one processor computation cycle.

(2) Offloading computation to the edge server

The total time of offloading computation of task *i* to the edge server ($T_{i,o-es}^{av}$) is equal to the sum of the time required to transmit the data of task *i* (Ds_i) to the edge server ($T_{i,\,t-es}^{av}$), plus the local computing time of the edge server (T_i^{l-es}) plus the time for sending back the results to the road authority vehicle ($T_{i,o-rs}^{av}$). $T_{i,o-es}^{av}$ is defined by:

$$\begin{aligned} T_{i,o-es}^{av} = &\quad T_{i,t-es}^{av} + T_i^{l-es} + T_{i,o-rs}^{av} \\ &= \left(\frac{Ds_i}{R_{\text{LTE-A/5G}}^{av}}\right) + \left(\frac{C_i}{F_{\text{CPU}}^{es}}\right) + \left(\frac{Rs_i}{R_{\text{LTE-A/5G}}^{es}}\right) \end{aligned} \tag{13}$$

Here ($R_{LTE-A/5G}^{av}$) represents the data transmission rate through the LTE-A/5 G interface of the road authority vehicle.

The cost of sending data to the edge server ($\Pi_{i,av}^{cell,o-es}$) is given by:

$$\Pi_{i,av}^{cell,o-es} = Ds_i \times \pi_{LTE-A}^{cell} \tag{14}$$

The computation cost of the edge server ($\Pi_{i,pv}^{cpt,o-es}$) is given by:

$$\Pi_{i,av}^{cpt,o-es} = C_i \times \pi_{es}^{cpt} \tag{15}$$

The energy consumed by the road authority vehicle to offload the data of task *i* to the edge server ($E_{i,o-es}^{av}$) is given by:

$$E_{i,o-es}^{av} = \vartheta \times (Ds_i \times e_{LTE-A/5G}^{av}) \tag{16}$$

Where $e_{LTE-A/5G}^{av}$ represents the energy consumed by the LTE-A/5 G interface to send one data unit to the Edge server.

(3) Sharing the computation with a nearby vehicle

The total time to share the computation with a nearby vehicle ($T_{i,s-nv}^{av}$) is equal to the sum of the local computation time of a percentage of task *i* by the road authority vehicle (T_i^{l-av}) plus the time required to transmit the task's data to the nearby vehicle ($T_{i,t-nv}^{av}$). We note δ the computation cycle ratio to be shared with the nearby vehicle of task *i*, such as $\delta = [0,1]$. This ratio is defined locally by the road authority vehicle according to it computational requirement. $T_{i,s-nv}^{av}$ is given by:

$$T_{i,s-nv}^{av} = T_i^{l-av} + T_{i,t-nv}^{av} = \left((1-\delta) \times \frac{C_i}{F_{CPU}^{av}}\right) + \left(\delta \times \frac{Ds_i}{R_{C-V2X}}\right) \tag{17}$$

The total energy consumption of sharing the computation with a nearby vehicle ($E_{i,s-nv}^{av}$) is equal to the sum of the energy of the local computation of a percentage of task *i* by the road authority vehicle (E_i^{l-av}) plus the energy consumed to transmit the task's data to the nearby vehicle ($E_{i,t-nv}^{av}$). We note by ϑ the type of road authority vehicle. $E_{i,s-nv}^{av}$ is given by:

$$E_{i,s-nv}^{av} = \vartheta \times \left(\left((1-\delta)\ \times C_i \times e_{CPU}^{av}\right) + \left(\delta \times Ds_i \times e_{C-V2X}^{av}\right)\right) \tag{18}$$

The shared computation of the task *i* by the nearby vehicle requires the payment of a computation cost ($\Pi_{i,av}^{cpt,s-nv}$), usually a monetary charge, by the road authority vehicle in the function of the calculation time, calculated using the number of processor cycles. $\Pi_{i,av}^{cpt,s-nv}$ is given by:

$$\Pi_{i,av}^{cpt,s-nv} = \gamma \times C_i \times \pi_{nv}^{cpt} \tag{19}$$

Here, π_{nv}^{cpt} refers to the remote computation cost of computing one task unit by the nearby vehicle.

4.3. Nearby vehicle computation model

When the nearby vehicle receives the shared task *i* from the road authority vehicle it can either accept or deny to compute the shared task based on the availability of its resources.

(1) Accept the computation

The total time to compute the shared task *i* ($T_{i,a}^{nv}$) is equal to the sum of the local computation time by the nearby vehicle (T_i^{l-nv}) plus the sending time of results to the road authority vehicle ($T_{i,s-rs}^{nv}$). $T_{i,a}^{nv}$ is given by:

$$T_{i,a}^{nv} = T_i^{l-nv} + T_i^{Rs} = \left(\delta \times \frac{C_i}{F_{CPU}^{nv}}\right) + \left(\delta \times \frac{Rs_i}{R_{C-V2X}}\right) \tag{20}$$

Here, F_{CPU}^{nv}represents the computation frequency of the nearby vehicle in processor cycles per second.

The total energy consumption to compute the shared task *i* ($E_{i,a}^{nv}$) is equal to the sum of the energy for local computation of task *i* by the nearby vehicle (E_i^{l-nv}) plus the energy consumed to send the results to the road authority vehicle ($E_{i,s-rs}^{nv}$). $E_{i,a}^{\text{nv}}$ is given by:

$$E_{i,a}^{\text{nv}}=\tau \times \left(E_i^{l-nv} + E_{i,s-rs}^{nv}\right)=\tau \times \left(\left(\ \delta \times C_i \times e_{\text{CPU}}^{nv}\right)+\left(\delta \times Rs_i \times e_{\text{C-V2X}}^{nv}\ \right)\right) \tag{21}$$

Here, e_{CPU}^{nv} represents the energy consumed by the nearby vehicle for one processor computation cycle, and e_{C-V2X}^{nv} represents the energy consumed by the C-V2X interface to send one data unit from the nearby vehicle to the road authority vehicle.

(2) Deny the computation

If the nearby vehicle denies sharing the task computation with the road authority vehicle, the time and energy of the shared computation will be zero, because the computation is performed entirely by the road authority vehicle.

4.4. Calculation of the link quality

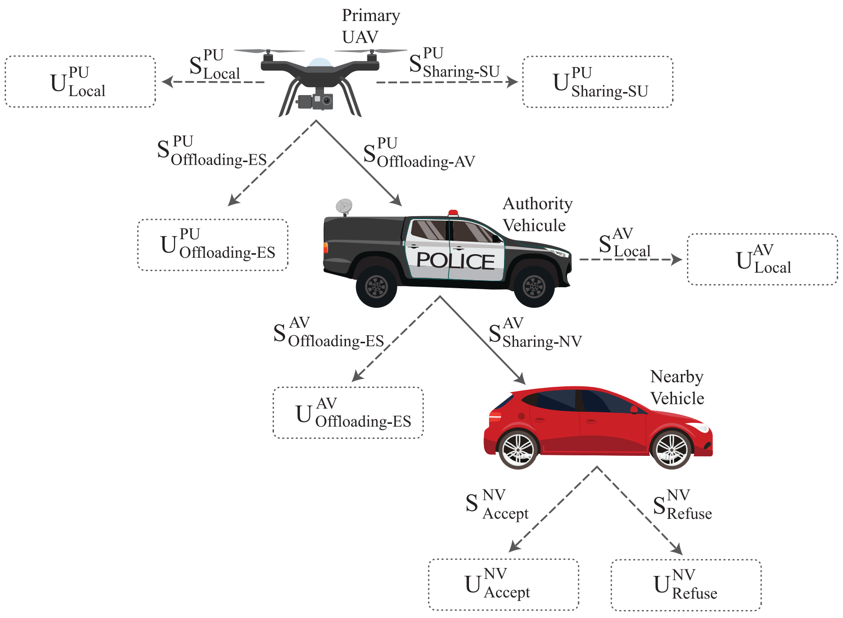
We model the link quality between two nodes *k* and *l* with the probability ($\phi_{k,l}^{sr}$). $\phi_{k,\,l}$ represents the probability that the packets will be successfully transmitted and received between these two nodes.$\phi_{k,l}^{sr}$ is defined as [35]:

$$\phi_{k,l}^{sr}(\tau) = \frac{N_{k,l}^{r}(t)}{N_{k,l}^{T}(t)} \tag{22}$$

Where $N_{k,l}^{r}(t)$ represents the number of packets successfully received during a communication interval τt, and $N_{k,l}^{T}(t)$ represents the total number of packets transmitted during the same communication interval.

$\phi_{k,l}^{sr}$ represents the probability that a sent packet will be successfully received between nodes *k* and *l*.$\phi_{k,l}^{sr}$ is defined as [35]:

$$\phi_{k,l}^{mr} = \frac{N_{k,l}^{r}(t)}{N_{k,l}^{m}(t)} \tag{23}$$



Primary UAV
S^{PU}_{Local}
U^{PU}_{Local}
$S^{PU}_{Sharing\text{-}SU}$
$U^{PU}_{Sharing\text{-}SU}$
$S^{PU}_{Offloading\text{-}ES}$
$S^{PU}_{Offloading\text{-}AV}$
$U^{PU}_{Offloading\text{-}ES}$
Authority Vehicule
POLICE
S^{AV}_{Local}
U^{AV}_{Local}
$S^{AV}_{Offloading\text{-}ES}$
$S^{AV}_{Sharing\text{-}NV}$
$U^{AV}_{Offloading\text{-}ES}$
Nearby Vehicle
S^{NV}_{Accept}
S^{NV}_{Refuse}
U^{NV}_{Accept}
U^{NV}_{Refuse}

Fig. 3. Synoptic representation of the utility and the strategies for each player.

Where $N^{r}_{k,l}$ represents the number of packets received during a communication interval τt, and $N^{m}_{k,l}$ is the number of packets that must be received during the same communication interval. From (22) and (23), the link quality $\phi_{k,\ l}$ between the node k and l is defined as [35]:

$$\phi_{k,l} = \phi^{sr}_{k,l}(t) \times \phi^{mr}_{k,l} \quad (24)$$

5. Sequential game for computation offloading/sharing for UAV-aided traffic monitoring

To solve the offloading/sharing computation decision-making problem described in sub-section 3.2, we adopt in this section an approach based on a dynamic sequential game. The main objective is to achieve the best balance between the computation delay, energy consumption, and communication/computation cost. We describe first the game formulation before investigating the existence of the Nash equilibrium (NE).

5.1. Game formulation

Game theory is considered as a powerful mathematical tool for studying decision-making problems when rational entities interact so that each of them can satisfy their own interests. In addition, several studies refer to game theory when studying decision-making process related to the computation offloading problems [2,5,6,25,28]. This led us to choose game theory as an enabling tool to model our decision-making problem, because of the decentralized nature of the decision-making process performed by network nodes in our system model and the complexity of designing a centralized decision-making algorithm.

We formulate our computational decision-making problem as a finite sequential game with perfect information, consisting of three players, namely: *(i)* the primary UAV, *(ii)* the road authority vehicle, and *(iii)* the nearby vehicle. In perfect information sequential game, players act sequentially, where each player makes strategic choices by observing the actions of the other players acting before it [28]. In our case, data is first collected by the primary UAV, the road authority vehicle only acts if the primary UAV decides to offload the computation tasks to it, and the nearby vehicle only acts if the road authority vehicle decides to share the computation tasks with it. The number of computation tasks is finite and is equal to the number of computation tasks necessary to proceed with the collected data. The game ends when the equilibrium is reached (all the system parameters are optimized for each collected data), or when the energy level of the drone becomes critical, *i.e.*, the minimum needed energy for the drone to return to the authority vehicle.

The sequential game is defined by *SG (N, S, U)*; where:

- $N = \{Primary\ UAV\ (PU),\ Authority\ Vehicle\ (AV),\ Nearby\ Vehicle\ (NV)\}$ represents the finite set of players;
- $S = \{S_{PU},\ S_{AV},\ S_{NV}\}$, represents the set of strategies for each player, respectively;
- U_i, represents the overall utility function of the system. The values of the global utility function are determined by the decisions made by the players and their respective utilities.

Fig. 3. shows a synoptic representation of the entire system.

The strategies and utilities of each player are detailed in the following.

5.2. Player's strategies

In our non-cooperative sequential game, each player is rational and aims to optimize its own utility. In the following, we describe the three players' strategies.

5.2.1. Primary UAV strategies

The primary UAV has three strategies (see Fig. 3):

(1) Local computation (S^{PU}_{Local}): in this strategy, all computation tasks are performed locally by the primary UAV.
(2) Sharing computation with secondary UAVs ($S^{PU}_{Sharing-SU}$): in this case, the drone shares the computation tasks with one or several UAVs of its nearby UAVs.
(3) Offloading computation to edge server ($S^{PU}_{Offloading-ES}$): in this strategy, the primary UAV offloads the computation tasks to be performed locally at the edge server.
(4) Offloading computation to the road authority vehicle ($S^{PU}_{Offloading-AV}$): in this strategy, the primary UAV offloads the computation tasks to be performed by the road authority vehicle.

5.2.2. Road authority vehicle strategies

The road authority vehicle has three strategies (see Fig. 3):

(1) Local computation (S^{AV}_{Local}): in this strategy, the road authority vehicle performs all computation tasks locally.

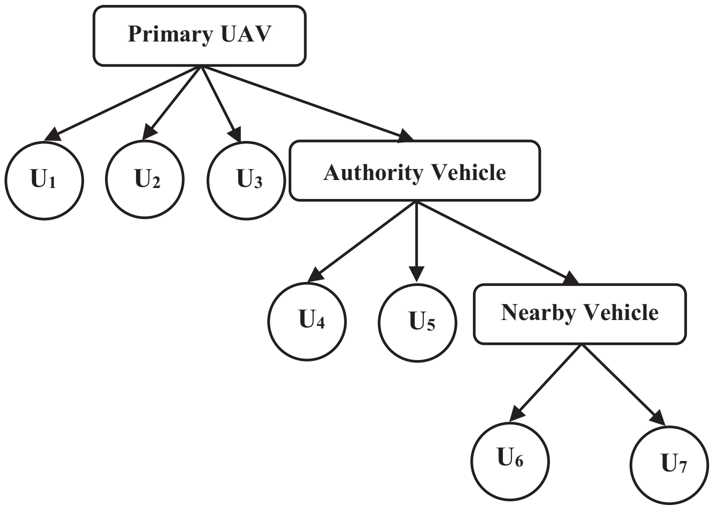


Fig. 4. Extensive representation of partial utilities.

Table 1
Strategic representation of partial utilities.

Authority Vehicle		S^{AV}_{Local}		$S^{AV}_{Offloading-ES}$		$S^{AV}_{Sharing-NV}$	
Nearby Vehicle		S^{NV}_{Accept}	S^{NV}_{Refuse}	S^{NV}_{Accept}	S^{NV}_{Refuse}	S^{NV}_{Accept}	S^{NV}_{Refuse}
Primary UAV	S^{PU}_{Local}	U$_1$					
	$S^{PU}_{Offloading-ES}$	U$_2$					
	$S^{PU}_{Sharing-SU}$	U$_3$					
	$S^{PU}_{Offloading-PV}$	U$_4$		U$_5$		U$_6$	U$_7$

(2) Offloading computation to an Edge server ($S^{AV}_{Offloading-ES}$): in this case, the road authority vehicle offloads the computation tasks to be performed by an edge server.
(3) Sharing computation with a nearby vehicle ($S^{AV}_{Sharing-NV}$): in this strategy, the road authority vehicle shares its computation tasks with a nearby vehicle.

5.2.3. Nearby vehicle strategies

The nearby vehicle has two strategies (see Fig. 3):

(1) Accept the computation (S^{NV}_{Accept}): in this strategy, the nearby vehicle accepts to share the computation with the road authority vehicle.
(2) Deny the computation (S^{NV}_{Deny}): in this strategy, the nearby vehicle denies to share the computation with the road authority vehicle.

5.3. Partial utilities

The player's utility function implements a correlation between a set of innovative system metrics: delay, energy, communication and computation cost. The partial system utility functions can be easily formulated to represent all the player's strategies involved in the computation decision-making process, see Fig. 4.

As shown in Fig. 4 and Table 1, seven possible strategic profiles can be listed reflecting the different possible computation scenarios, which are presented below:

$$S_1 = \{S^{PU}_{Local}, *, *\}$$
$$S_2 = \left\{S^{PU}_{Offloading-ES}, *, *\right\}$$
$$S_3 = \left\{S^{PU}_{Sharing-SU}, *, *\right\}$$
$$S_4 = \left\{S^{PU}_{Offloading-AV}, S^{AV}_{Local}, \ *\right\}$$
$$S_5 = \left\{S^{PU}_{Offloading-AV}, S^{AV}_{Offloading-ES}, *\right\}$$
$$S_6 = \left\{S^{PU}_{Offloading-AV}, S^{AV}_{Sharing-NV}, \ S^{NV}_{Accept}\right\}$$
$$S_7 = \left\{S^{PU}_{Offloading-AV}, S^{PV}_{Sharing-NV}, \ S^{NV}_{Deny}\right\} \quad (25)$$

The corresponding partial utility functions for each strategy profile are detailed in Fig. 4 and Table 1. For example, U_1 represents the partial utility of primary UAV corresponding to local computation strategy. For that, only computation delay and energy are considered. The computation and communication costs are omitted because the UAV calculates itself at local all tasks and uses only free communication technology (*i.e.*, C-V2X). The detailed utilities are given as follows:

$$U_1 = \begin{cases} \alpha\left(T^{pu}_{i,l}\right) \\ \beta\left(E^{pu}_{i,l}\right) \end{cases}$$

$$U_2 = \begin{cases} \alpha\left(T^{pu,o-es}_{i}\right) \\ \beta\left(E^{pu,o-es}_{i}\right) \\ \gamma\left(\Pi^{cell,o-es}_{i,pu}\right) \\ \lambda\left(\Pi^{cpt,o-es}_{i,pu}\right) \\ \delta\left(\phi_{pu,es}\right) \end{cases}$$

$$U_3 = \begin{cases} \alpha\left(T^{pu}_{i,s-su}\right) \\ \beta\left(E^{pu}_{i,s-su}\right) \\ \gamma\left(\Pi^{cell,s-su}_{i,pu}\right) \\ \lambda\left(\Pi^{cpt,s-es}_{i,pu}\right) \\ \delta\left(\phi_{pu,su}\right) \end{cases}$$

$$U_4 = \begin{cases} \alpha\left(T^{pu}_{i,o-av} + T^{av}_{i,l}\right) \\ \beta\left(E^{pu}_{i,o-pv} + E^{av}_{i,l}\right) \end{cases}$$

$$U_5 = \begin{cases} \alpha\left(T^{pu}_{i,o-av} + T^{av}_{i,o-es}\right) \\ \beta\left(E^{pu}_{i,o-av} + E^{av}_{i,o-es}\right) \\ \gamma\left(\Pi^{cell,s-es}_{i,av}\right) \\ \lambda\left(\Pi^{cpt,o-ev}_{i,av}\right) \\ \delta\left(\phi_{pu,av} \times \phi_{av,es}\right) \end{cases}$$

$$U_6 = \begin{cases} \alpha\left(T^{pu}_{i,o-av} + T^{av}_{i,s-nv} + T^{nv}_{a,i}\right) \\ \beta\left(E^{pu}_{i,o-av} + E^{av}_{i,s-nv} + E^{nv}_{a,i}\right) \\ \lambda\left(\Pi^{cpt,s-nv}_{i,av}\right) \\ \delta\left(\phi_{pu,av} \times \phi_{av,nv}\right) \end{cases}$$

$$U_7 = U_4 = \begin{cases} \alpha\left(T^{pu}_{i,o-av} + T^{av}_{i,l}\right) \\ \beta\left(E^{pu}_{i,o-av} + E^{av}_{i,l}\right) \end{cases} \quad (26)$$

5.4. Global utility function

The context of our study is related to a road traffic monitoring scenario where a fleet of UAVs provides aerial assistance to road authority vehicles. The data collected by the UAVs must be processed as soon as possible in order to extract relevant information about the traffic condition. The road authority vehicles use this information to make an efficient intervention at the right time, particularly for the dangerous and urgent situation on the road. All these reasons, make the computation of the corresponding tasks, a delay-sensitive operation regarding the computation time response. In addition, UAVs have limited energy capabilities, so optimizing the computation energy consumption is essential to save the UAV's battery life-time and extend the monitoring mission duration. Moreover, the transmission of task data using cellular networks often requires to pay a transmission cost related to the size of the transmitted data [2]. Furthermore, computational services provided by the edge server and nearby vehicles also inflicts to pay a computation cost [28]. Moreover, the energy and the delay of transmitting a task's data through a wireless link can be seriously affected by the quality of this link. The wireless communications are intermittent and unreliable by nature, and they more challenging in scenarios involving drones because of the weather perturbators (*e.g.*, rain, wind, temperature, etc.) and the drone's own features (*e.g.*, altitude, velocity, etc.). In fact, using a link with poor quality consumes extra energy and requires more time for data transmission. The link quality is usually expressed by the probability of successful packet transmission in a small time interval [30].

Taking in consideration all these system metrics, we propose a new system utility function for the computation decision-making of task *i* in the form of a joint equation of a set of innovative metrics: the computation delay, the energy consumption, the cellular communication cost, the remote computation cost, and the link quality. The system utility function is given as:

$$U_i = \frac{\alpha \times T_i + \beta \times E_i}{1 + \delta \times \phi_{i,j}} + \gamma \times \Pi_i^{cell} + \lambda \times \Pi_i^{Cpt} \quad (27)$$

Where for each task *i*, T_i represents the computation time, E_i represents the computing energy consumption, Π_i^{cell} represents the cost of cellular communication, Π_i^{Cpt} represents the cost of remote computation, and $\phi_{i,j}$ represents the quality of the link between vehicle *i* and *j*. $\alpha, \beta, \gamma, \lambda$ and δ represent the weight parameters for the computation time, energy, cellular communication cost, remote computation cost, and link quality, respectively. We consider $\alpha, \beta, \gamma, \delta, \lambda \in [0,1]$ and $\alpha + \beta + \gamma + \delta + \lambda = 1$. These weighting factors offer much greater flexibility in modeling a wide range of vehicular realistic situations with different specific requirements. Therefore, depending on the intended scenario or even the current status of the system, different tasks may have different weighting parameters.

Since we aim to minimize the system utility, the proposed utility function is defined as a monotonic function that exactly follows the rise or the decline of the system metrics: delay, energy and the communication/computation cost. Moreover, as we aim to maximize the link quality and minimize the system utility, the utility function increases for a bad link quality and decreases when the link quality is good.

The utility function (27) is composed of disjointed variables that are delay, energy, cellular communication cost, and computation cost. Therefore, to calculate this function, these variables must be normalized as follows:

$$T_i = \frac{T_i - minT}{maxT - minT};\ E_i = \frac{E_i - minE}{maxE - minE};\ \Pi_i^{cell} = \frac{\Pi_i^{cell} - min\Pi^{cell}}{max\Pi_i^{cell} - min\Pi^{cell}};$$

$$\Pi_i^{Cpt} = \frac{\Pi_i^{Cpt} - min\Pi^{Cpt}}{max\Pi^{Cpt} - min\Pi^{Cpt}}$$

Where, *maxT, maxE*, $max\Pi_i^{cell}$and $max\Pi^{Cpt}$represent the maximum values of delay, energy, cellular communication cost, and remote computation cost, respectively. *minT, minE*, $min\Pi_i^{cell}$ and $min\Pi^{Cpt}$ represent the minimum values of delay, energy, cellular communication cost, and remote computation cost, respectively. This normalization makes all variables in the interval [0.1]. The reason we have not standardized the link quality is that this variable is a probability that is already between 0 and 1.

5.5. Nash equilibrium

In this sub-section, we investigate the existence of the Nash equilibrium (NE) for the proposed sequential game. A strategy profile is a Nash equilibrium if each strategy taken by a player represents the best response to the strategies of other players. In other words, a Nash equilibrium is a combination of strategies such that no player can achieve better utility by unilaterally changing its decision [31]. This optimal and satisfactory solution would finally be achieved in order to obtain the best possible performances.

In the proposed three-player sequential game, NE represents a satisfactory decision profile while for each computation task correspond the best decision strategy that optimizes conjointly all the system metrics (delay, energy, and cost).

Theorem 1. "*Every finite extensive-form game with perfect information has a pure-strategy Nash equilibrium*" *[28]*.

Proof: The proof of Theorem 1 is given in [32].

Lemma 1. *SG (N, S, U) is a finite sequential game with perfect information.*

Proof: see the previous sub-section 5.1.
Theorem 1 implies that the proposed finite sequential game with perfect information (see subsection 5.1) has at least one Nash equilibrium that can be derived following a finite number of iterations.

Nash equilibrium is achieved when players find the decision profile that ensures a minimum value for the global utility of the system. The optimal decision profile d_i^* which admits a Nash equilibrium for a task *i* is defined as:

$$d_i^* = \underset{T,E,\Pi^{cell},\Pi^{cpt},\phi}{\arg\min}\ U_i \quad (28)$$

To compute and define the system metrics optimal value (the optimal value of the computation delay, energy, and cost) in NE for each computation task, we propose a decentralized algorithm that is based on the algorithm in our previous work [5].

As described in Fig. 5, the basic idea of our algorithm consists that in each computation decision iteration corresponding to the computation of task *i*, the players start by initializing the decision strategy to local computation before evaluating their respective system metrics in the real-time networks state. Then, each player proceeds by selecting and comparing its current initial strategy with the other possible computation strategies to define the best strategy that optimizes the system metrics. If it finds a better strategy that minimizes its utility, the player will choose this new strategy as the best decision strategy, otherwise, it maintains the current strategy as the best decision strategy. Therefore, the players continue testing the possible strategy until reaching the NE state where the archived decision strategies cannot be more enhanced. Hence, when an equilibrium state is reached, each player will execute its corresponding optimal strategy $S^* = \{S^{PU*}, S^{AV*}, S^{NV*}\}$. In addition, according to Theorem 1 presented above, our algorithm would achieve equilibrium in a finite number of iterations. Although the algorithm would be executed simultaneously by all players, the decision-making process follows a sequential order. To implement our proposed algorithm, we assumed that the players exchange short messages to inform each others of the decision made by each player.

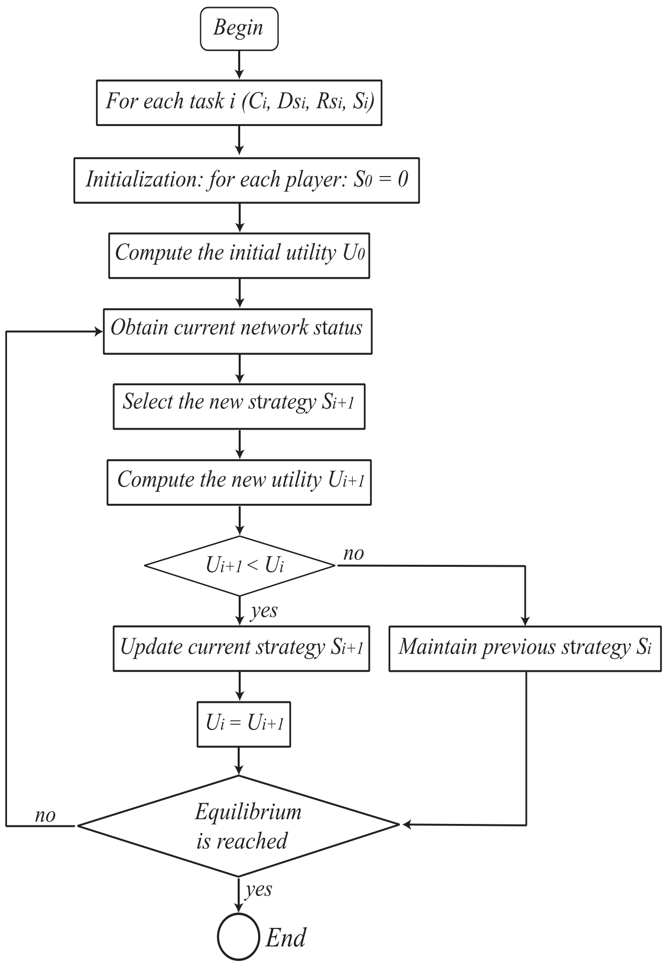

Begin
For each task i (Ci, Dsi, Rsi, Si)
Initialization: for each player: S0 = 0
Compute the initial utility U0
Obtain current network status
Select the new strategy Si+1
Compute the new utility Ui+1
Ui+1 < Ui
no
yes
Update current strategy Si+1
Maintain previous strategy Si
Ui = Ui+1
Equilibrium is reached
no
yes
End

Fig. 5. Computation Offloading/Sharing Algorithm flowchart.

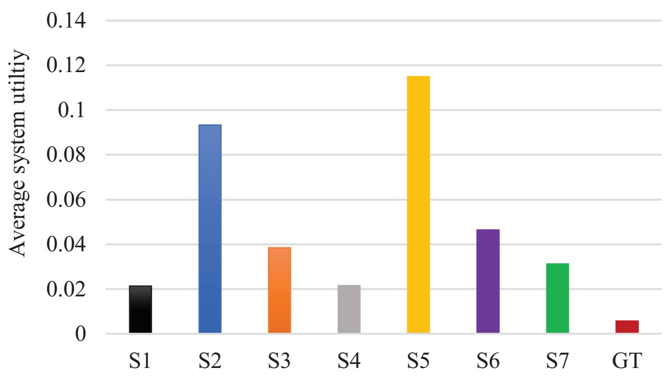


Fig. 6. Average System Utility.

Table 2
Simulation parameters.

Parameters	Values
# UAV	3
Tasks C_i	[10,..., 100] (x10^6)
D_i	[10,..., 200] (x10^3)
Rs_i	[5,..., 15]
$(\alpha, \beta, \gamma, \delta, \lambda)$	(1/5, 1/5, 1/5, 1/5, 1/5)
F^{pu}_{CPU}	12 GHZ
$(e^{pu}_{CPU}, e^{es}_{CPU}, {}^{ev}_{CPU})$	(1,0, 3) units
(R_{LTE-A}, R_{C-V2X})	50,4 MB/s
π^{cell}_{LTE-A}	1×10^{-6} units
π^{cpt}_{es}	1×10^{-6} units
π^{cpt}_{nv}	5×10^{-7} units
$e^{pu}_{\mathrm{LTE-A}} = e^{pu}_{\mathrm{C-V2X}}$	800 units

6. Numerical results

In this section, we present the simulation scenarios to evaluate the performance of our proposed game theoretical model and discuss the numerical results.

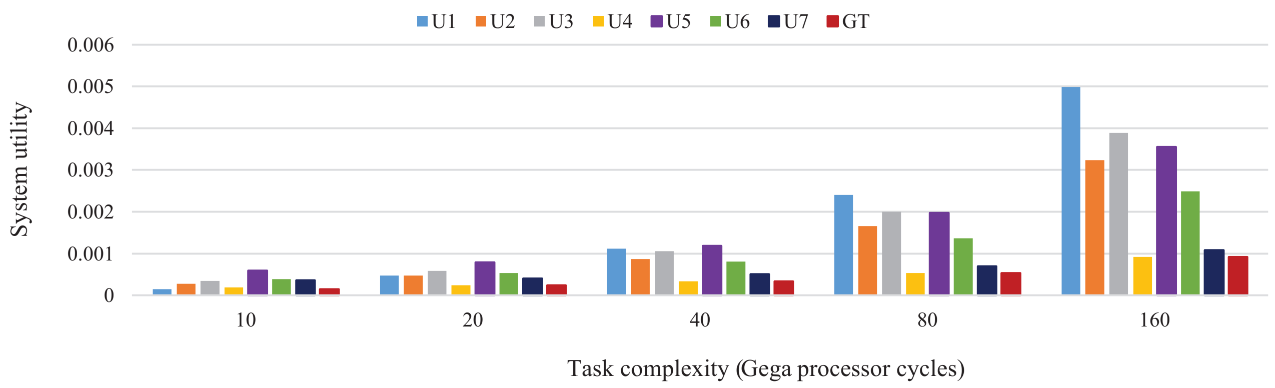
For the various experimentations, we consider that the processor capability of the road authority vehicle (F^{av}_{CPU}) is five times faster than that of the UAV (F^{pu}_{CPU}), and 1.6 times faster than that of nearby vehicles (F^{nv}_{CPU}). The processor capability of the edge server (F^{es}_{CPU}) is ten times faster than that of authority vehicles [2, 5, 25]. We consider the computation capabilities of a Skydio 2 drone such as an example of nowadays existing drone that mount Jetson TX2 GPU (Graphical Processing Unit) with a capacity up to 12 GHZ. As assumed in [2, 5, 25], and [36], we also consider that the energy consumed by UAV to send a data unit via the LTE-A interface ($e^{pu}_{\mathrm{LTE-A}}$) and the C-V2X interface ($e^{pu}_{\mathrm{C-V2X}}$) is 800 - 1000 times more than the energy consumed for local computation (e^{pu}_{CPU}) of the same data unit by the UAV. The energy consumed by the local computation of a data unit on an electric vehicle (e^{ev}_{CPU}) is three times more than the energy consumed by the local computation (e^{pu}_{CPU}) of the same data unit on the UAV since electric vehicles are three times more powerful than the drone. Without loss of generality and for simplicity concern, in our experimentation, we consider only gasoline vehicles except in the experimentation in Fig. 8.b. It should be noted that our algorithm still accepting any other scenario including electric vehicles by setting the variable ϑ=0. We set the transmission rate of the LTE-A interface (R_{LTE-A}) and the C-V2X Interface (R_{C-V2X}) to 50.4 MB/s [7]. Besides, we consider that sending data via the LTE-A interface (π^{cell}_{LTE-A}) consumes 1×10^{-6} units [3], whereas sending data via the C-V2X interface is free of charge. Furthermore, we assume that the calculation performed in the edge server (π^{cpt}_{es}) consume 1×10^{-6} units, and the calculations performed in a nearby vehicle (π^{cpt}_{nv}) consume 5×10^{-7} units. Finally, as in similar existing works in [2, 5] and [25], we assigned initially equal importance to system metrics. Therefore, we set an equal value to the weighting factors (α=β=γ=δ=λ=1/5). Afterward, we evaluate the impact of the variation of these weighting factor values on the system performance. The source code used in the experiments is available as open-source in GitHub [41].

The main simulation parameters are summarized in Table 2.

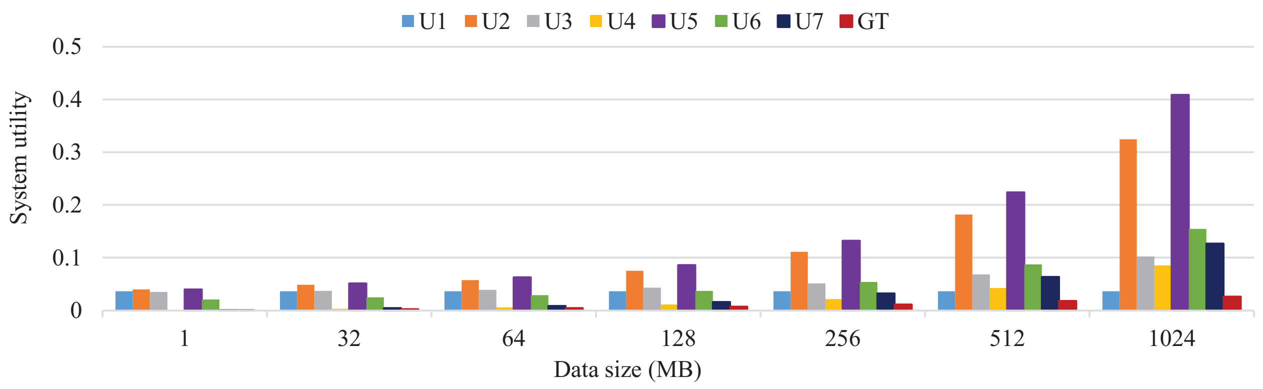
In the rest of this section, we provide a comparative study between the proposed three-player sequential game approach and five basic computation strategies, namely: *(i)* local computing by the UAV, *(ii)* offloading computation to edge server, *(iii)* sharing computation with UAVs, *(iv)* offloading computation to the road authority vehicle, and *(v)* sharing computation with nearby vehicles. In the previous computation scenarios, the players choose a computation strategy without any intelligence. In addition, the seven strategies listed in the previous section are taken into account. In the first model S_1, the computation tasks are performed by the primary UAV. Whereas, in the third strategy S_3, the computation tasks are shared with secondary UAVs. Furthermore, in the second and fifth strategies (S_2 and S_5), the edge server performs the computation tasks. In the fourth and seventh strategies (S_4 and S_7), road authority vehicle executes the computation tasks locally. Whereas in the sixth strategy S_6, the computation tasks are shared with the nearby vehicle.

In the evaluation in Fig. 6, we study the average system utility for different data processing scenarios. We compare our strategic system utility of our game theoretical offloading/sharing computation (GT) with that of basic system utilities, where the system's nodes adopt without any intelligence one of the computation strategies previously described in sub-section 5.3.

From the results in Fig. 6, we can remark clearly the effectiveness of our proposed sequential game model (GT) compared to the other



(a) Impact of task's complexity on the average system utility.



(**b**) Impact of task's data size on the average system utility.

Fig. 7. Impact of the variation of the task characteristics (complexity and data size) on the average system utility. (a) Impact of task's complexity on the average system utility. (b) Impact of task's data size on the average system utility.

seven computation strategies in terms of average system utility. Indeed, our computation strategy based on the sequential game-theoretical approach is the most efficient and offers the optimal system utility. This is because, at each task computation, our proposed offloading/sharing computation approach uses the decision profile obtained in NE to make strategic decisions. Effectively, GT balances between different computation strategies to determine the most efficient strategy that optimizes the system utility and ensures the best tradeoff between energy efficiency, computation delay, communication cost, and computation cost.

In Fig. 7, we investigate the impact of the calculation complexity and data size of the computational task on the average system utility. We start in Fig. 7.(a) by studying the impact of the task's computation complexity on the average system utility. In this evaluation, we fix the size of the data (D_i) and we vary each time the task complexity (C_i). Then, we study in Fig. 7.(b), the impact of the size of the task's data on the system average utility. In this evaluation, we fix the task complexity (C_i) and vary each time the task's data size (D_i).

We remark from the results in Fig. 7.(a) and Fig. 7.(b) that local computation on the primary UAV is the most efficient strategy for less complex computational tasks with heavy data size. This is because sending big data via the costly cellular communication needs more delay, consumes plus energy and requires more communication cost. Inversely, we remark that sharing and offloading computation strategies are more appropriate for medium and very complex tasks with small or medium data sizes, respectively. The main justification is due to the powerful capabilities that offer the edge server and the advantage of distributed collaborative computation. The communication/computation cost is generally compensated by the rapid computation and energy economization. We observe from Fig. 7.(b) that the average system utility for all computation scenarios increases as the task's data size increases. This augmentation is principally due to the energy and delay overhead caused by the primary UAV for sending the data using a wireless interface. We also note that the data size has no impact on the average system utility when calculating the task locally on primary UAV. Because in this kind of scenario, primary UAV will only send the calculation results to the authority vehicle. The size of the computation results is generally small and needs only small energy and delay to be sent back. We remark as well, the effectiveness of our sequential game approach (GT), which always converges to the most effiecient computation strategy thanks to the computation profile achieved in NE that gives the optimal average system utility, regardless of the task's complexity and data size.

In Fig. 8, we assess the flexibility of our utility function in regard to the characteristics of the computational task through the variation in the weighting parameter values. In this evaluation, we focused on studying the variation of delay and energy weighting factors noted α and β respectively, since the main concern of our work is principally optimizing computation delay and energy consumption that represent the most critical metrics of our system. In this experimentation, we fix the communication and computation cost weighting factor values and we vary that of delay and energy. We consider also when studying the impact of energy consumption, two scenarios including the case of an electric vehicle and that of a gasoline vehicle.

From Fig. 8.(a), we can remark that when the delay weighting factor is equal to 0.2, the proposed sequential game approach does not have the best delay average, because delay does not carry much weight in the computation decision regarding the others system metrics. By increasing the delay weighting factor to 0.5 or more, we remark that the proposed sequential game approach provides the best delay average. Indeed, our game-theoretical model always chooses the best strategy that gives the smallest system utility that corresponds to the optimal delay with a high weighting factor. Similarly, we remark from in Fig. 8.(b) that by increasing the energy weighting factor, our game theoretical

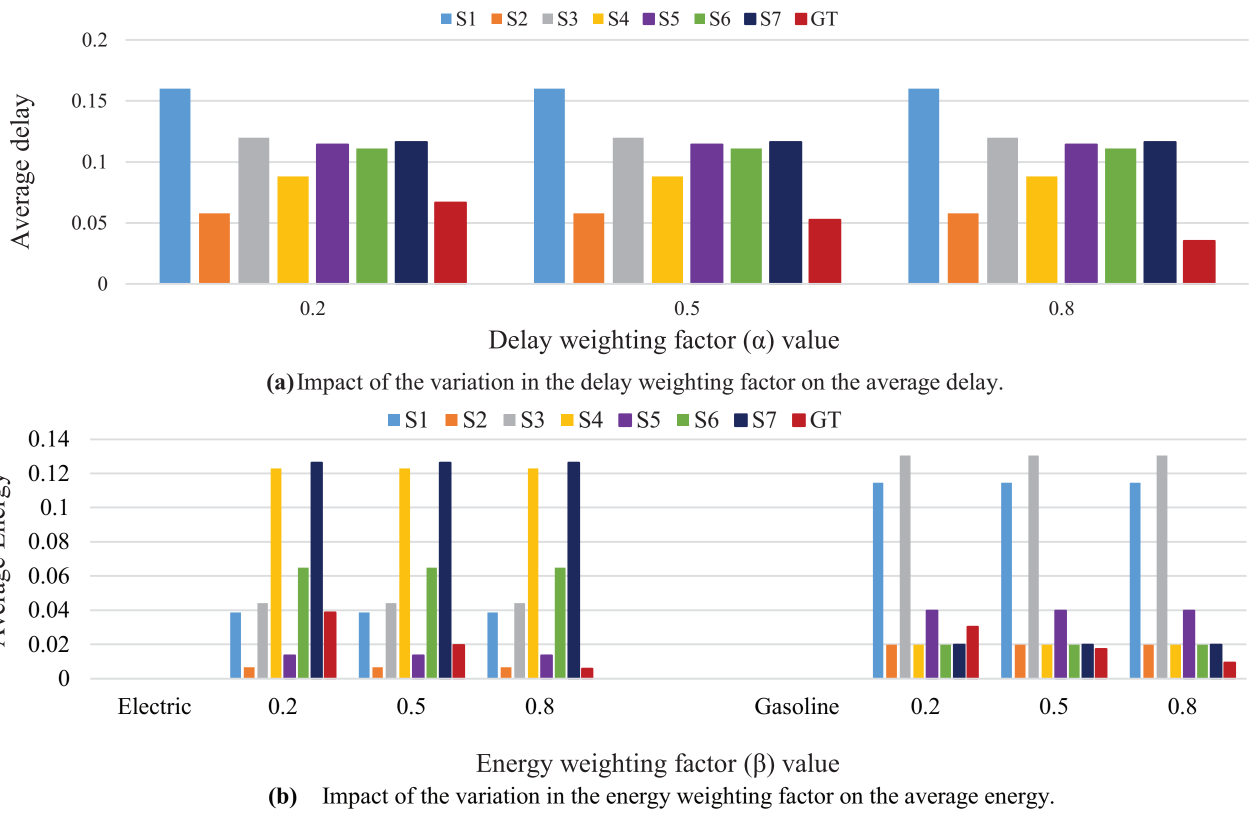


Fig. 8. Impact of the variation in the system main weighting factors (delay and energy) on system metrics. (a) Impact of the variation in the delay weighting factor on the average delay. (b) Impact of the variation in the energy weighting factor on the average energy.

approach obtains a better energy average. As a conclusion from this experimentation, we confirm our initial assertion which stipulates varying the system weighting factors allow our proposed model to flexibly support various characteristics of different computation tasks and therefore different user's applications.

In Fig. 9, we perform a comparative study between our proposed computation algorithm based on the sequential game and the related computation algorithm DOSC (Distributed Offloading Sharing Computation) proposed in [5] based on the average energy consumption in Fig. 9.(a) and the average delay in Fig. 9.(b). For the comparison scenario, we run the two algorithms under the same parameters for different simulation scenarios. In each scenario, we fix the task's complexity and we vary each time the size of the task's data. Then, we compare the performance of our algorithm and that of DOSC algorithm with respect to the average energy consumption in Fig. 9.(a) and also with respect of average delay in Fig. 9.(b).

From Fig. 9, we can see that our computation algorithm based on the sequential game approach (GT) is more efficient than the DOSC algorithm both in terms of average energy consumption as we can clearly see in Fig. 9.(a) and also in terms of average delay as shown in Fig. 9.(b). This is because our proposed computation algorithm based on the sequential game considers more advanced computation possibilities while DOSC considers only two computation strategies, *i.e.*, offload intensive tasks to be performed by the authority vehicle and this latter can share the computation with one of it nearby vehicles. In addition to the above-cited strategies, our algorithm offers more advanced computation strategies to better improve the system utility and optimize the system metrics, *i.e.*, the possibility to offload the computation to a more powerful edge server and the possibility of sharing the computation with nearby UAVs. By combining these innovative computation strategies and considering the system delay and energy, in addition of the computation/communication cost and the quality of link. This in-

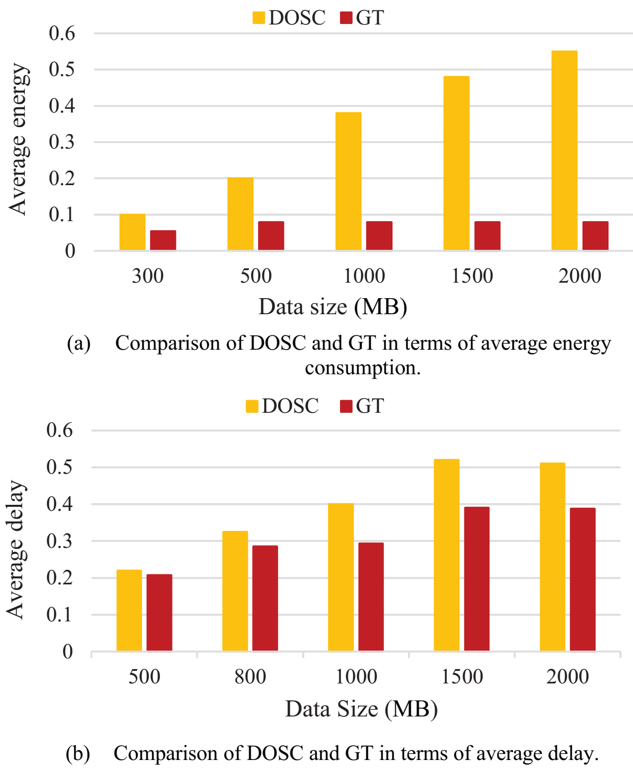


Fig. 9. Comparison of the performance of our game theoretical approch (GT) and that of the related work DOSC [5]. (a) Comparison of DOSC and GT in terms of average energy consumption. (b) Comparison of DOSC and GT in terms of average delay.

novative system utility is used to achieve the best balance between the different system metrics. Whereas the DOSC algorithm considers a more simple system utility composed from only the energy and delay as main parameters to make strategic decisions.

7. Conclusion

In this paper, we consider the problem of implementing an effective approach for processing the data collected by UAV in a multi UAV-aided road traffic monitoring scenario, which involves computation offloading/sharing decision-making problems. The main purpose is to decrease computation delay while optimizing the energy overhead as well as computation/communication cost. We first start by proposing a novel system architecture that enables computation offloading and sharing. Then, we define a novel system utility function that combines computation delay, energy overhead, quality of the link, as well as communication and computation cost. Furthermore, we formulate the offloading/sharing decision-making problem through a theoretical game approach as a three-player sequential game and then we study the existence of Nash equilibrium. After that, we design a computation algorithm to reach such an equilibrium. The simulation results showed that our model, based on the sequential game, outperforms other baseline computation approaches by offering better performance in terms of overall system utility with an efficiency that varies between 43% and 97% depending on the computation approach, and allows a better average computation time and energy consumption.

As future perspectives to our work, we intend to improve the system model by offering new alternatives for offloading/sharing computation and take into consideration more advanced network parameters such as the drone hovering and reception energy. We hope also performing more advanced experimentation to evaluate the impact of the economic cost of communication and computation on the performance of the system utility.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ahmed Alioua: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Supervision. **Houssem-eddine Djeghri:** Methodology, Software, Formal analysis, Writing - original draft. **Mohammed Elyazid Tayeb Cherif:** Methodology, Software, Formal analysis, Writing - original draft. **Sidi-Mohammed Senouci:** Validation, Writing - review & editing. **Hichem Sedjelmaci:** Validation, Writing - review & editing.

References

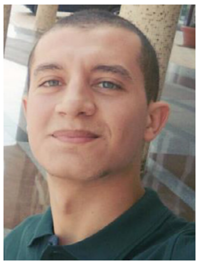
[1] H. Shakhatreh, Unmanned Aerial Vehicles (UAVs): a survey on civil applications and key research challenges, In IEEE Access 7 (2019) 48572–48634.
[2] M. Messous, S. Senouci, H. Sedjelmaci, S. Cherkaoui, A game theory based efficient computation offloading in an UAV network, in IEEE Trans. Veh. Technol. 68 (5) (2019) 4964–4974 May.
[3] H. Ghazzai, H. Menouar and A. Kadri, "On the Placement of UAV Docking Stations for Future Intelligent Transportation Systems," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, 2017, pp. 1–6.
[4] X. Fan, C. Huang, B. Fu, S. Wen, X. Chen, UAV-assisted data dissemination in delay–constrained VANETs, Mobile Inf. Syst. 2018 (2018) 1–12.
[5] Alioua, et al., Efficient data processing in software-defined UAV-assisted vehicular networks: a sequential game approach, In Wirel. Pers. Commun. 4 (101) (2018) 2255–2286.
[6] M. Messous, A. Arfaoui, A. Alioua and S. Senouci, "A sequential game approach for computation-offloading in an UAV network," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1–7.
[7] A. Ashraf Ateya, A. Muthanna, R. Kirichek, M. Hammoudeh, A. Koucheryavy, Energy- and latency-aware hybrid offloading algorithm for UAVs, in IEEE Access 7 (2019) 37587–37600.
[8] H. Seliem, R. Shahidi, M.H. Ahmed, M.S. Shehata, Drone-based highway-VANET and DAS service, in IEEE Access 6 (2018) 20125–20137.
[9] O.S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, A. Rachedi, UAV-assisted supporting services connectivity in urban VANETs, in IEEE Trans. Veh. Technol. 68 (4) (2019) 3944–3951 April.
[10] O.S. Oubbati, A. Lakas, M. Guneş, F. Zhou, M.B. Yagoubi, UAV-assisted reactive routing for urban VANETs, in: in Symposium on Applied Computing, ACM, 2017, pp. 651–653.
[11] O.S. Oubbati, A. Lakas, N. Lagraa, M.B. Yagoubi, UVAR: an intersection UAV-assisted vanet routing protocol, in: In Wireless Communications and Networking Conference (WCNC), IEEE, 2016.
[12] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, G. Jeon, Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: electronic eyes in the skies to spot clogged roads, In IEEE Consum. Electron. Mag. 8 (3) (2019) 81–86 May.
[13] Z. Shafiq, R. Abbas, M.H. Zafar, M. Basheri, Analysis and evaluation of random access transmission for UAV-assisted vehicular-to-infrastructure communications, in IEEE Access 7 (2019) 12427–12440.
[14] R. Zhang, F. Zeng, X. Cheng and L. Yang, "UAV-Aided Data Dissemination Protocol with Dynamic Trajectory Scheduling in VANETs," 2019 IEEE International Conference On Communications (ICC), Shanghai, China, 2019.
[15] O.S. Oubbati, A. Lakas, P. Lorenz, M. Atiquzzaman, A. Jamalipour, Leveraging communicating UAVs for emergency vehicle guidance in urban areas, 2019 Early access.
[16] S. Ortiz, C.T. Calafate, J. Cano, P. Manzoni, C.K. Toh, A UAV-based content delivery architecture for rural areas and future smart cities, in IEEE Internet Comput. 23 (1) (2019) 29–36.
[17] M. Khabbaz, J. Antoun, C. Assi, Modeling and Performance Analysis of UAV-Assisted Vehicular Networks, in IEEE Trans. Veh. Technol. 68 (9) (2019) 8384–8396.
[18] H. Sedjelmaci, M.A. Messous, S.M. Senouci, I.H. Brahmi, Toward a lightweight and efficient UAV-aided VANET, Trans. Emerg. Telecommun. Technol. 30 (8) (2018).
[19] S.A. Hadiwardoyo, C.T. Calafate, J. Cano, Y. Ji, E. Hernández-Orallo and P. Manzoni, "Evaluating UAV-to-Car Communications Performance: from Testbed to Simulation Experiments," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2019, pp. 1–6.
[20] M.A. Khan, W. Ectors, T. Bellemans, D. Janssens, G Wets, Unmanned aerial vehicle-based traffic analysis: a case study for shockwave identification and flow parameters estimation, Remote Sens. 10 (3) (2018) 458.
[21] J. Zhu, S. Chen, W. Tu, K. Sun, Tracking and simulating pedestrian movements at intersections using unmanned aerial vehicles, Remote Sens. (Basel) 11 (8) (2019) 925.
[22] H. Ghazzai, H. Menouar and A. Kadri, "On the placement of UAV docking stations for future intelligent transportation systems," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, 2017, pp. 1–6.
[23] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides and C.G. Panayiotou, "Optimized vision-directed deployment of UAVs for rapid traffic monitoring," 2018 IEEE International Conference On Consumer Electronics (ICCE), Las Vegas, NV, 2018, pp. 1–6.
[24] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi and L.A. Saidane, "Monitoring road traffic with a UAV-based system," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1–6.
[25] M. Messous, H. Sedjelmaci, N. Houari and S. Senouci, "Computation offloading game for an UAV network in mobile edge computing," 2017 IEEE International Conference On Communications (ICC), Paris, 2017, pp. 1–6.
[26] D. Callegaro and M. Levorato, "Optimal computation offloading in edge-assisted UAV systems," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1–6.
[27] H. Menouar, et al., UAV-enabled intelligent transportation systems for the smart city: applications and challenges, IEEE Commun. Mag. 55 (3) (2017) 22–28.
[28] M. Liwang, J. Wang, Z. Gao, X. Du, M. Guizani, Game theory based opportunistic computation offloading in cloud-enabled IoV, in IEEE Access 7 (2019) 32551–32561.
[29] J. Wang, Y. Shao, Y. Ge, R. Yu, A survey of vehicle to everything (v2x) testing, in Sensors 19 (2) (2019) 334.
[30] H. Zhou, H. Wang, X. Chen, X. Li, S. Xu, Data offloading techniques through vehicular ad hoc networks: a survey, IEEE Access 6 (2018) 65250–65259.
[31] J. Li, G. Kendall, R John, Computing nash equilibria and evolutionarily stable states of evolutionary games, IEEE Trans. Evol. Comput. 20 (3) (2015).
[32] H.W. Kuhn, and A.W. Tucker, "Contributions to the theory of games," Princeton University Press, Vol. 2, 1953.
[33] M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, J. Gozalvez, Analytical models of the performance of C-V2X Mode 4 vehicular communications, n IEEE Trans. Veh. Technol. 68 (2) (Feb. 2019) 1155–1166.
[34] H. Sedjelmaci, A. Boudguiga, I. Ben Jemaa, S.M. Senouci, An efficient cyber defense framework for UAV-Edge computing network", Ad Hoc Netw. 94 (2019).
[35] E.B. Smida, S.G. Fantar and H. Youssef, "Video streaming forwarding in a smart city's VANET," in 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), pp 88–95, 2018.
[36] Stetsko, L. Folkman and V. Matyaš, "Neighbor-based intrusion detection for wireless sensor networks," 2010 6th International Conference On Wireless and Mobile Communications, Valencia, 2010, pp. 420–425.
[37] J. Liu, Q. Zhang, Code-partitioning offloading schemes in mobile edge computing for augmented reality, in IEEE Access 7 (2019) 11222–11236.
[38] X. Chen, Decentralized computation offloading game for mobile cloud computing, IEEE Trans. Parallel Distrib. Syst. 26 (4) (2015) 974–983.

[39] X. Hou, Z. Ren, W. Cheng, C. Chen and H. Zhang, "Fog based computation offloading for swarm of drones," ICC 2019 - 2019 IEEE International Conference On Communications (ICC), Shanghai, China, 2019, pp. 1–7.

[40] Asheralieva, D. Niyato, Hierarchical Game-Theoretic and Reinforcement Learning Framework for Computational Offloading in UAV-Enabled Mobile Edge Computing Networks With Multiple Service Providers, in IEEE IoT J. 6 (5) (2019) 8753–8769.

[41] https://github.com/HoussemDjeghri/UAVs-for-Traffic-Monitoring-A-Sequential-Game-based-Computation-Offloading-Sharing-Approach.

Ahmed Alioua (M'19) received the Ph.D. degree in computer science from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2019. From 2014 to 2019, he was an assistant professor/researcher at the department of computer science of Constantine 2 University, Constantine, Algeria. He is an assistant professor/researcher at the department of computer science of Mohamed Seddik Benyahia University, Jijel, Algeria, since October 2019. His-main research interests include internet of vehicles, unmanned aerial vehicles, Software-Defined Networking (SDN), game theory and BlockChain.


Houssem-eddine Djeghri received the Master degree in computer science from University of Constantine 2, Constantine, Algeria, in 2019. His-main research interests include vehicular networks and unmanned aerial vehicles.

Mohammed Elyazid Tayeb Cherif received the Master degree in computer science from University of Constantine 2, Constantine, Algeria, in 2019. His-main research interests include vehicular networks, unmanned aerial vehicles, and game theory.

Sidi Mohammed Senouci (M'06) received the Ph.D. degree in computer science from the University of Paris 6, Paris, France, in October 2003. Since September 2010, he has been a Full Professor with Institut Supérieur de l'Automobile et des Transports, a major French postgraduate school located in Nevers, France, and a component of the University of Burgundy. He holds seven international patents on these topics and published his work in major IEEE conferences and renowned journals. His-research interests include vehicular communications, ad hoc and sensor networks, Transmission Control Protocol over wireless, wireless and mesh networks, cooperative networks, and performance evaluation. Prof. Senouci is a member of the IEEE Communications Society (ComSoc) and an Expert Senior of the French Society of Electricity and Electronics (SEE). He has been serving as a Technical Program Committee (TPC) member of the following International Federation for Information Processing, Association for Computing Machinery, or IEEE conferences and workshops (ICC, GLOBECOM, PIMRC, GIIS, VTC, WiVeC, MWCN, IWWAN, Wire- less Days, WITS, etc.). He is the Chair of the IEEE ComSoc Information Infrastructure and Networking Technical Committee (2014–2016). He was a Cochair of the Ad Hoc and Sensor Networking Symposium in the 2011 IEEE Global Communications Conference (GLOBECOM) and a Cochair of the Next Generation Networking Symposium in the 2012 IEEE International Conference on Communications. He was a Vice-Chair of the Selected Areas in Communications Symposium in the 2010 IEEE Globecom, a Cochair of the Vehicular Technology Conference Symposium in the 2010 IEEE Wireless Communications and Mobile Computing Conference, and a TPC Cochair of the VehiCom2009 Workshop. He was the founding Chair of the Ubiroads2007 work-shop. He was the Guest Editor of a special issue of the UBICC journal and was the Special Track Cochair in the 2008 International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) on intelligent transportation systems. He is a founding Coeditor of the IEEE ComSoc Ad Hoc and Sensor Network Technical Committee Newsletter.

Hichem Sedjelmaci (M'14) received the Ph.D. degree in telecommunication systems from University of Tlemcen, Algeria, in 2013. From 2013 to 2016, he was a postdoctoral researcher with the DRIVE Laboratory, University of Burgundy, Nevers, France. In 2017, he was a Research Engineer in cyber security at the Institute of Technological Research SystemX. In 2018, he joined Orange Labs as a Senior Research Engineer in cyber security and artificial intelligence. He published his work in major IEEE conferences and premium journals (IEEE Transactions).