Computers, Materials & Continua
DOI: 10.32604/cmc.2023.030242
Article

Resource Management in UAV Enabled MEC Networks

Muhammad Abrar[1], Ziyad M. Almohaimeed[2,*], Ushna Ajmal[1], Rizwan Akram[2], Rooha Masroor[3] and Muhammad Majid Hussain[4]

[1]Bahauddin Zakariya University, Department of Electrical Engineering, Multan, 60000, Pakistan
[2]Department of Electrical Engineering, College of Engineering, Qassim University, Buraidah, 51452, Saudi Arabia
[3]COMSATS University WAH Campus, Islamabad, 47040, Pakistan
[4]Department of Electrical and Electronics Engineering, University of South Wales, Pontypirdd, CF37 1DL, UK
*Corresponding Author: Ziyad M. Almohaimeed. Email: z.mohaimeed@qu.edu.sa
Received: 22 March 2022; Accepted: 07 May 2022

Abstract: Mobile edge cloud networks can be used to offload computationally intensive tasks from Internet of Things (IoT) devices to nearby mobile edge servers, thereby lowering energy consumption and response time for ground mobile users or IoT devices. Integration of Unmanned Aerial Vehicles (UAVs) and the mobile edge computing (MEC) server will significantly benefit small, battery-powered, and energy-constrained devices in 5G and future wireless networks. We address the problem of maximising computation efficiency in U-MEC networks by optimising the user association and offloading indicator (OI), the computational capacity (CC), the power consumption, the time duration, and the optimal location planning simultaneously. It is possible to assign some heavy tasks to the UAV for faster processing and small ones to the mobile users (MUs) locally. This paper utilizes the k-means clustering algorithm, the interior point method, and the conjugate gradient method to iteratively solve the non-convex multi-objective resource allocation problem. According to simulation results, both local and offloading schemes give optimal solution.

Keywords: Mobile edge computing; internet of things; UAVs; ground mobile users

1 Introduction

Smartphones, smart appliances, and sensors are examples of "resource-constrained" IoT [1] devices due to their small size and limited storage, computational, and energy resources. These concentrated on-board operations eventually lead to greater energy consumption, which slows down and introduces latency into devices that are currently utilised for computation-intensive applications like augmented reality (AR), virtual reality (VR), pattern recognition, and monitoring. Due to limited resources and the increasing number of applications and volume of mobile traffic on IoT devices [2], resource allocation is a significant challenge i.e., minimising energy consumption, increasing computation efficiency, increasing computation bits, decreasing costs, decreasing completion time, etc.

The unmanned aerial vehicles, also known as remotely piloted drones (automatic or manual), have a wide variety of applications in both the military and civilian sectors [3,4]. The UAVs are used in wireless communications because they can adjust their altitude, mobility, and adaptability. By reducing the shadowing and blockage effects, obstacles can be avoided. Small uncrewed aerial vehicles provide cost-effective and energy-efficient solutions for mobile ground users spread across a large geographic area by establishing reliable connections with low transmit power. Due to their portability and ability to reach areas where traditional wired infrastructure is impractical, UAVs are frequently used for emergencies. UAV can serve as an aerial base station named a UAV-assisted wireless systemthat can be used to provide reliable network capacity and communication to ground users. It can also act as a flying mobile user referred to as a "cellular-connected" UAV, as it enables reliable and low-latency communication within the cellular network, as in real-time video streaming. Two broad categories based on their hardware are Fixed and Rotary wings. Fixed-wing unmanned aerial vehicles are small planes with fixed wings. They are not slowed by heavy weights; they are lightning fast. They cannot just stand still and stay in the air. They must keep going. Rotary-wing unmanned aerial vehicles are those that have rotating wings. It is difficult for them to move, and it is quite heavy. They are capable of both static and free flight in the air [5].

The European Telecommunications Standards Institute's (ETSI) Industry Specification Group (ISG) introduced the Mobile Edge Computing (MEC) concept for 5G networks in 2014. The research carried out in [6] defines MEC as: It provides an IT service environment and cloud computing capability close to a radio access network (RAN) mobile devices. "Edge Computing" refers to "a diverse set of techniques for relocating computing and storage tasks away from remote clouds (public or private) and closer to the data source" [7]. According to this theory, end users' mobile devices can access more computing power. MEC is an excellent and promising solution for resource-constrained mobile devices and heavy computation tasks at the edge nodes, such as base stations (BSs) or user devices with high computation capabilities. MEC is composed of two components: mobile users and the server. Mobile devices offload computation tasks to a powerful mobile edge computing cloud (MEC server/edge server) in order to comply with quality of service (QoS) and quality of experience (QoE) standards while also conserving energy, reducing latency, and increasing device processing speed.

In areas where fixed terrestrial MEC networks are unavailable or destroyed by natural disasters, Intelligence Edge Servers (IESs) can provide mobile edge services and on-demand computation resources. Utilizing a MEC system equipped with a UAV has several advantages, including the following: By enabling a broader range of applications, the LoS links in the U-MEC system increase the system's flexibility and efficiency [8]. U-MEC enhances computing services by increasing system capacity and providing high-bandwidth access to users, i.e., wide coverage [9]. It is a low-latency, low-energy offloading system that significantly improves the overall system's performance [10]. The assisted MEC architecture for ground users depicted in Fig. 1, which also functions as an aerial MEC server-enabled base station enabling the ground users to delegate computationally intensive tasks to one or more UAVs. Due to unforeseen events or natural disasters, ground-based stations may be unable to provide services in areas with sparse or non- existent terrestrial infrastructure. Devices that must comply with QoS requirements can use this to significantly reduce their energy consumption [11].

Mobile devices benefit from increased battery life, lower latency, and improved computational performance when their radio resources are used efficiently through U-MEC architecture. This is accomplished by users offloading computationally intensive tasks to a nearby MEC server, thereby improving the overall computation performance of the system. The MEC server is attached to UAVs, which offer greater flexibility, are easier to deploy, and are mobile, all of which contribute to increased radio coverage. They are frequently used in areas devoid of terrestrial infrastructure. Reduced energy consumption, increased computation efficiency (measured in computation bits per joule of energy), increased computational bits, shorter completion time (measured as the difference between local computing time and offloading computation time), and cost reduction are all achieved in U-MEC systems through resource management.

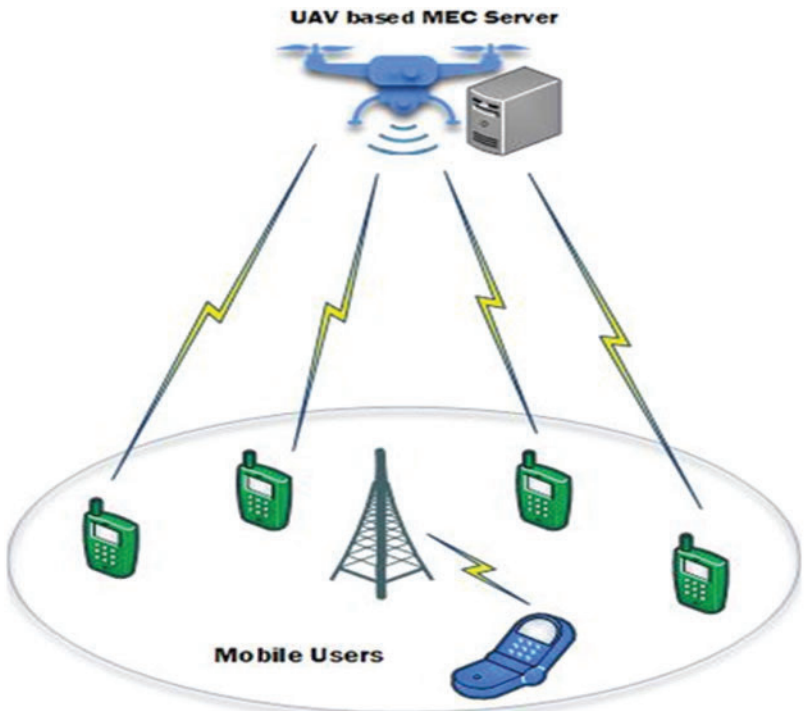

Figure 1: UAV equipped with MEC server

Research carried on in [12] suggests that a problem of maximising computational efficiency can be solved by taking into account central processing unit (CPU) frequencies, the user's maximum energy consumption, the position of the UAVs, and the user's transmit power constraints, and jointly optimising transmit power and offloading time for users. The Lagrangian Duality Method is used to calculate the transmitting power and CPU frequency, while the sequential convex approximation (SCA) technique is used to solve the UAV's path problem. In [13], energy consumption and computation bits are increased by optimising user association, trajectory scheduling, and resource allocation simultaneously to achieve maximum computation efficiency under local CPU frequency allocation. After that, the optimization problem is solved iteratively using a double loop structure [14]. discusses a multi-UAV MEC system with multiple ground users. The overall energy consumption of the system is expected to be significantly reduced. The allocation of resources, task scheduling, deployment, offloading decision, location, and number of UAVs operating under time constraints are optimised using a two-layered optimization method. The author of [15] discusses how to extend the battery life of both users and UAVs by reducing energy consumption during computation tasks. By considering task offloading, uplink and downlink bit allocation, and trajectory design, it is possible to formulate a joint optimization problem for the energy budget and latency of a UAV. To solve this problem, SCA and block coordinate descent (BCD) techniques are combined in a novel optimization algorithm. Reduced energy consumption and user delay are objectivesof optimising communication and computing resources, task di- vision decisions, and UAV placement [16]. An SCA method is used to solve this non-convex problem. By optimising hover time, resource allocation, and task scheduling in conjunction with ground users' QoS requirements and UAV computing resources, UAVs consume less total energy in [17]. The iterative BCD algorithm is used to find suboptimal solutions to solve this joint optimization problem. UAVs consume less energy in [18]. This energy consumption includes that required for communication, computation, and flight. Computational and communication resources are allocated based on the UAV's trajectory design and the computation bits allocated within a given time slot. The issue is resolved using a combination of SCA and the Lagrangian duality method. In [19], the UAV's energy consumption optimization within the constraints of computation bit size and energy harvesting causality is accomplished by jointly considering offloading computation bits, user CPU frequency, and the UAV's trajectory. An alternative algorithm based on the SCA method is proposed for this purpose. Because of the UAV's limited power, the author describes an energy-efficient algorithm in [20] that employs a three-layered computation offloading strategy. The position of the UAVs can be dynamically adjusted using the UAV position optimization algorithm to provide the best possible service to all users. The tasks that users delegate to UAVs are predicted using an long short-term memory (LSTM) task prediction algorithm. To maximise energy efficiency, the system employs a task-offloading strategy. This optimization problem is solved by optimising the frequency, offloading bits, active user transmit power, and trajectory of UAV to minimise energy consumption is described in [21]. This problem incorporates constraints on computing tasks, harvesting, and an energy storage system. This objective is accomplished through the use of SCA and design and artificial intligence (DAI) algorithms. The objective of [22] is to minimise energy consumed in computation processing and the time required to complete UAV detection in wind farms. The frequency of UAV computation, offloading power, modes, and time are all optimised simultaneously while maintaining wind turbine (WT) accuracy, UAV flight speed and transmission power constraints, and computation frequency constraints. To mitigate the wind's effect on WTs, multi- sortie detection trajectory planning and UAV scheduling (DTPUS) have been proposed. The iterative offloading trajectory and computation offloading (OTCO) algorithm are then used to optimise both the computation offloading and the inspection trajectory's path. When Lagrangian duality is used, the speed of UAV computation, offloading, and power calculation are all significantly improved.

Research performed in [23] aims to reduce the total amount of energy consumed by users in the air-ground integrated MEC network due to the limited power capacity of IoT devices. To ensure that UAV latency and power consumption are limited, as well as bandwidth utilisation and computational capacity, the author devised an integrated optimization problem involving uplink power control and channel allocation, user association, and 3D positioning of UAVs. This issue is resolved using an effective optimization algorithm based on the BCD method. Work performed in [24] proposes a strategy for users to conserve energy by optimising power allocation, re-source partitioning, uplink, and downlink bit allocation, the number of processed bits at the user-power UAV's allocation and scheduling, the trajectory of the user-UAV subject to resource partitioning fractions, bit causality in uplink and downlink, and the UAV's initial/final location and maximum speed. The SCA method is used to address the issues of power allocation, resource partitioning, bit allocation in the uplink and downlink, and the trajectory of the UAV. Reference [25] describes a time division multiple access (TDMA) based MEC server problem of optimising a UAVs trajectory that obtains both local and global optimal solutions. The global optimum solution is found through a two-dimensional search of all possible UAV locations. Simultaneous optimization of slot allocation and task partitioning is performed using an augmented Lagrangian active method. The maximum energy consumption by users in the NOMA-based UAV-assisted MEC system is minimised by defining a joint optimization problem (min-max problem) [26]. By optimising task data, computing resource allocation and UAVs trajectory under users' task delays, total task data, mobility, and UAV trajectory constraints, a comprehensive strategy is used to reduce energy consumption across all ground users. The location of a fixed point can be optimised and pinpointed with the assistance of the fixed point service (FPS). MEC equipped with a UAV makes use of a TDMA- based model to increase the UAV's energy efficiency and service time in [27]. Energy consumption can be reduced by optimising resource allocation to users, the duration of UAV hovering, and the order in which users are associated. This issue is resolved using the BCD method. Author in [28] models enables energy-efficient resource allocation and trajectory design in UAV-mounted cloudlet. This results in a reduction in energy consumption and improved computing services. Communication and computation resources, the UAV trajectory, user transmit power, and computation load allocation are jointly optimised with subject to user offloading, the UAV's energy budget for communication, computing capabilities, and mechanical operations. SCA and Dinkelbach algorithms are used to accomplish the task at hand. Reference [29] investigates resource allocation in a wirelessly powered MEC system using a UAV. The goal of all ground users in binary or partial offloading modes is to optimise offloading time, user transmit power, and CPU frequencies while limiting the UAV's speed and energy harvest. In both partial and binary offloading, optimised algorithms are used to maximise the number of computation bits simultaneously. Reference [30] priorities energy-aware resource allocation in order to maximise utility across the MEC system's vehicle social networks (SIoV). Vehicles transmit energy in accordance with the law of energy consumption evolution at any given point in time. In cooperative and non-cooperative situations, dynamic programming is used to optimise the dynamic power allocation of vehicles with fixed UAV trajectory. With acceptable distance of user-UAV and vehicle components that have been offloaded from the UAV, a search algorithm is used. In this paper, multi-objective resource allocation problem is formulated by using multiple UAVs to maximise the computation efficiency of ground mobile users. The individuals listed below made the greatest contributions to the paper's objectives.

(i) For both local execution and offloading, we defined the computation efficiency maximisation problem under latency, power and QoS requirements by jointly optimising the offloading/user association indicator, computation capacity, power consumption, time duration and optimal location Z.
(ii) This non-linear resource allocation problem can be efficiently solved by combining an unsupervised learning algorithm with an interior point method, which solves the problem iteratively and aids in the discovery of the optimal solution to achieve the desired goal.

The rest of the paper is organised as follows: The system model and problem formulation are introduced in this Sections 2 and 3 respectively. Section 4 delves into the algorithm that was used to solve the problem at hand. Section 5 summarizes and discusses the simulation results. The paper is finally concluded in Section 6.

2 System Model

System model is depicted in Fig. 2. In 3D euclidean coordinates, an U-MEC network is taken into consideration with 'B' multiple mobile ground users (GUs) and 'A' rotary-winged multiple UAVs, hovering at fixed altitude 'H'. The whole process is categorized into three steps. In first step, heavy tasks like face recognition, live video streaming, traffic or environment monitoring, augmented/virtual reality that are to be executed are offloaded by the GUs to at least one and only one unique UAV flying above the GU at a specific distance. Executing the GU's task by the UAV-MEC server having sufficient computation capability and resources takes place in the second step. For a predetermined period of time, the UAV remains connected to the offloading GU within its coverage area. The third step involves the processed GU tasks' results like face identification, rendering the video streams, traffic or environment analysis which are sent back to the GUs. In the end, GUs download those results.

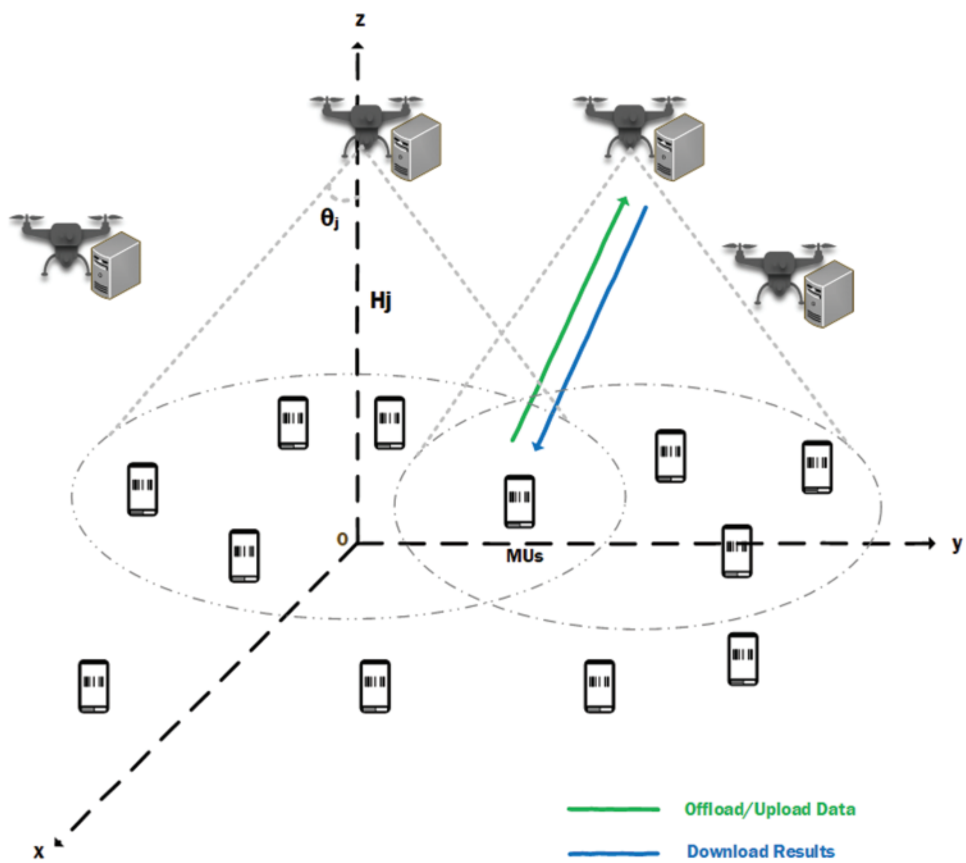

Figure 2: UAV-MEC network

Line-of-sight (LoS) paths between UAVs and GUs are considered because UAVs fly higher than MUs and have a low probability of encountering scatterers. While small MUs tasks are performed locally, larger and more complex tasks are delegated to the UAV, resulting in a more efficient system. While offloading computation tasks, the TDMA scheme is used to avoid co-channel interference. For example, UAVs and other radio users can share a frequency channel due to the TDMA protocol, which divides signals into time slots to minimise interference. Numerous GUs can share a single radio frequency channel by utilising only a fraction of its capacity. Tab. 1 depicts inputs and symbols used in the this paper.

Table 1: Notations used

Symbol	Definition	Symbol	Definition
A	No. of a UAVs i.e., 20	$C_{b,\,max}$	Max. battery consumed by GU
B	No. of b GUs i.e., 150	C_{max}	Max. energy available i.e., 20 dBm
A	Set of UAVs {1, 2, ... ,A}	t_{ab}	Offloading transmission rate of UAV assigned to the GU
B	Set of GUs {1, 2, ... ,B}	$(x_b, y_b, 0)$	Coordinates of GUs
u_{ab}	Offloading indicator/User association variable of GU i.e., 0 or 1	(X_a, Y_a, H_a)	Coordinates of UAVs with fixed altitude Ha
W	System's Bandwidth i.e., 1 MHz	H_a	UAVs Altitude i.e., min = 10, max = 20
l	Same time block 'l' required for the executable tasks i.e., 6 ms	θ_a	UAVs antenna half power beamwidth i.e., min = π /6, π /3
Q_b	Min. data bits transferred i.e., 50	$H_a \tan\theta_a$	Coverage region of UAV
c_b	GUs Computation Capacity	s_{ab}	Horizontal distance between the GU and UAV i.e., 500 m
$c_{j,\,max}$	Max. computation capacity of GU i.e., 1 GHz	d	Channel power gain outside the antenna's bandwidth
c_{ab}	Maximum computational capacity of UAV to the associated GU	d_o	Channel power gain at 1 m reference distance i.e., 1.42^{e-4}
L_{ab}	Time duration between the UAV and GU	d_{ab}	Uplink channel gain between GU and UAV
P^t_{ab}	Transmission/Execution power of the GU assigned to the UAV	β_b, v_b, w	CPU model positive constants i.e., $10 - 2^{e9} + 1$, 1^{e3}, 2
C_b	Constant energy consumed by the GU i.e., 0.00001W	$K_{b,\,max}$	Max. allowed GUs associated to the UAV

3 Mathematical Model

The model in Fig. 2 is considered that shows clusters of A UAVs: $\mathcal{A} = \{1, 2, \ldots, A\}$ and B GUs: $\mathcal{B} = \{1,2,\ldots,B\}$. The ground user performs smaller tasks locally, while larger tasks are delegated to a nearby UAV. This decision is made by the offloading/user association indicator variable u_{ab}:

$$u_{ab} = \{0, 1\}, \quad \forall b \in \ B, \forall a \in \ A u_{ab} = \{0, 1\}, \qquad \forall b \in \ \mathcal{B}, \forall a \in \ \mathcal{A} \tag{1}$$

- If user b conducts the task locally, then local execution takes place : $u_{ab}/u_{0b} = 1, \forall a = 0$.
- If user b does not upload the task to the UAV a: $u_{ab} = 0, \forall a = 0$
- If the user uploads the task to the UAV, offloading takes place: $u_{ab} = 1, \forall a \neq 0$

The downloading time for results from the UAV to the GU is so smallthat it is negligible. The transmission power p^t_{ab} is:

$$p^t_{ab} = \frac{\text{The total CPU task/cycles}}{\text{Computational capacity of the GU } c_b} \tag{2}$$

The horizontal distance between the GU and the UAV is:

$$s_{ab} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}, \qquad \forall b \in \mathcal{B}, \ \forall a \in \mathcal{A} \tag{3}$$

Data rate during offloading from the GU to the UAV is:

$$t_{ab} = W\log_2 \left(1 + \frac{d_{ab}p^t_{ab}}{\theta_a^2 \left(H_a^2 + s_{ab}^2\right)}\right), \qquad \forall b \in \mathcal{B}, \ \forall a \in \mathcal{A} \tag{4}$$

During offloading, the time duration allotted to the GU with UAV is denoted by L_{ab}.. The coordinates of UAVs (X_a, Y_a, H_a) at elevated angle θ_a are set close to the coordinates of GUs $(x_b, y_b, 0)$ with horizontal spacing s_{ab} between the UAV and the GUs, for the optima location Z. Each UAV is deployed with a directional antenna whose beamwidth can be adjusted accordingly. With both the azimuth θ and elevation ψ angles, the antenna gain can be calculated as:

$$D = \begin{cases} \dfrac{d_o}{\theta_a^2} & if 0 \le \theta \le \theta_a and 0 \le \psi \le \theta_a \\ d \approx 0 & otherwise, \end{cases} \tag{5}$$

d can be set to 0 for simplicity. If the GUs are assumed to be located outdoors, then LoS link will be made between the GU and UAV, but in urban areas, due to the presence of obstacles, a probabilistic LOS channel model is used, which is given by:

$$p_{LOS} = \frac{1}{1 + ie^{(-j[\theta a - i])}} \tag{6}$$

where i and j are the environmental-dependent channel parameters. For the channel model, TDMA and block fading models are assumed. Each time slot has the same channel in the block fading model. All-time slots of length ‘l’ are faded in the same way. While in the TDMA model, multiple GUs share the same frequency channel, which is divided into distinct time slots to minimise interference between both the GUs and UAVs.

3.1 Problem Statement

A metric known as computational efficiency is used to measure the U-MEC system’s energy efficiency in bits/Joule, which is given by:

$$\text{Computational Efficiency} = \frac{\text{Total Computed Bits}}{\text{Consumed Energy}} \tag{7}$$

Calculating the number of computed bits per Joule allows one to determine the efficiency of a system. This paper uses the least amount of energy and more computed bits possible as computation efficiency is a trade-off between computed bits achieved and energy consumption. An efficient algorithm is used to maximise the number of computed bits per Joule of energy consumed. Joint optimization of decision variables such as offloading/user association indicator u_{ab}, computation capacity c_b, power consumption p^t_{ab}, time duration L_{ab} and optimized location Z (X_a, Y_a, H_a, θ_a) takes place under latency and QoS constraints for both local execution (no offloading) and offloading.

3.2 Objective Function

$$\max_{U,C,P,L,Z} \frac{\sum_{b=1}^{B}\left(\frac{lc_b}{v_b}+\sum_{a=1}^{A}u_{ab}W\log_2\left(1+\frac{s_{ab}p_{ab}^{t}}{\theta^2\left(H_a^2+s_{ab}^2\right)}L_{ab}\right)\right.}{\sum_{b=1}^{B}\left(C_{b,max}+\beta_b c_j^w t+\sum_{a=1}^{A}p_{ab}^{t}L_{ij}\right)} \tag{8}$$

constrained by:

$$C0: u_{ab} \le 1, \quad \forall b \tag{9}$$

$$C1: \sum_{b=1}^{B} u_{ab} \le K_{b,max}, \quad \forall b \tag{10}$$

$$C2: lc_b v_b + \sum_{a=1}^{A} u_{ab}W\log_2\left(1+\frac{s_{ab}p_{ab}^{t}}{\theta^2\left(H_a^2+s_{ab}^2\right)}L_{ab} \ge Q_b, \forall b \tag{11}$$

$$C3: C_b + \beta_b c_b^w l + \sum_{a=1}^{A} p_{ab}^{t}L_{ab} \le C_{max}, \quad \forall b \tag{12}$$

$$C4: c_b \le c_{b,max}, \quad \forall b \tag{13}$$

$$C5: \sum_{b=1}^{B} L_{ab} \le l, \quad \forall a \tag{14}$$

$$C6: L_{ab} \le u_{ab}l, \quad \forall a, b \tag{15}$$

$$C7: p_{ab}^{t} \le u_{ab}C_{b,max}, \quad \forall a, b \tag{16}$$

$$C8: u_{ab}s_{ab} \le H_a tan\theta_a, \quad \forall a, b \tag{17}$$

$$C9: u_{ab} = \{0,1\}, \quad \forall a, \ b \tag{18}$$

$$X_a^{min} \le X_a \le X_a^{max}, \quad \forall a \tag{19}$$

$$Y_a^{min} \le Y_a \le Y_a^{max}, \quad \forall a \tag{20}$$

$$H_a^{min} \le H_a \le H_a^{max}, \quad \forall a \tag{21}$$

$$\theta_a^{min} \le \theta_a \le \theta_a^{max}, \quad \forall a \tag{22}$$

The numerator of the objective function comprises of the computation efficiency while the energy consumption during no offloading and offloading is the denominator, where $\mathbf{U} = \{u_{ab}\}_{b\in\mathcal{B},a\in\mathcal{A},}$, $\mathbf{C} = \{c_{ab}\}_{b\in\mathcal{B},a\in\mathcal{A},}$, $\mathbf{P} = \{p_{ab}^{t}\}_{b\in\mathcal{B},a\in\mathcal{A},}$, $\mathbf{L} = \{L_{ab}\}_{b\in\mathcal{B},a\in\mathcal{A},}$, $\mathbf{Z} = \{X_a, Y_a, H_a, \theta_a\}_{a\in\mathcal{A}}$. To ensure that each task is only performed once, C0 specifies that the MU either completes it locally or offloads it to a specific UAV. In C1, a single UAV cannot have more than the maximum number of GUs connected to it. C2 is a QoS constraint that applies when the total number of data bits used (for both local execution and offloading) exceeds the minimum number of data bits required to complete the task. The total energy consumption by the GU is less than or equal to the maximum amount of energy that can be used to complete the task, as specified in C3. In C4, GU's computation capacity must be less than or equal to the allowed maximum computation capacity of the GU. When a task is offloaded and then executed, its duration must be less than or equal to that of the 'l' time block. C6 denotes the amount of time it takes to from the UAV and MU to establish communication i.e., $o_{ij} = 1$. C7 is total GUs power consumption, which should always be less than or equal to the GUs maximum available power. UAVs must be within the coverage area of the GU to whom the GU has delegated the task, according to C8. C9 is a binary offloading indicator variable i.e., 0 or 1. Execution is performed locally if the $u_{ab} = 0$ is zero. There is an offloading procedure if the $u_{ab} = 1$, When $u_{ab} = \{0, 1\}$, the min. and max. limit for the UAVs horizontal coordinates (X), vertical coordinates (Y), altitude (H) and the half beamwidth/elevation angle (θ) of antenna must lie in the feasible region.

4 Proposed Algorithm

The given non-linear and non-convex multi-objective resource allocation complex problem constrained by non-linear constraints is solved by utilizing efficient algorithm which involves unsupervised learning algorithm and interior point method. It eventually helps in finding the optimal solution by decreasing the complexity of the system. The proposed algorithm's pseudocode is elaborated in Fig. 3 and it consists of three stages.

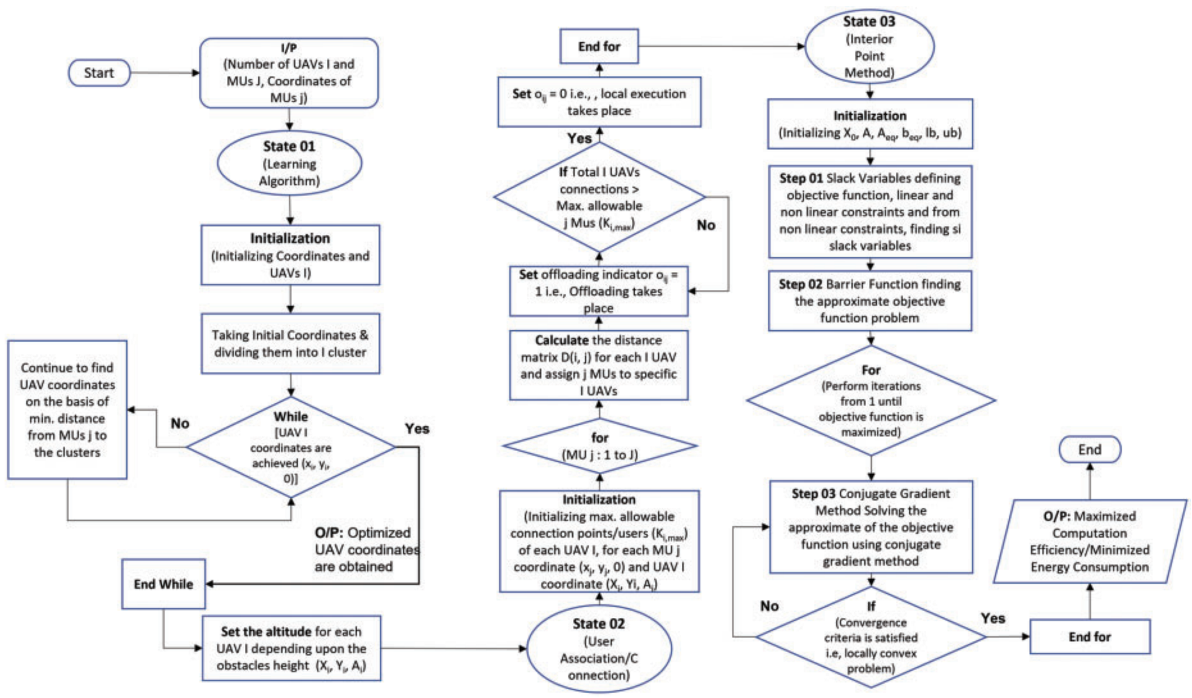

Figure 3: Pseudocode for proposed algorithm

The K-means Clustering Algorithm is used in the first stage. At this point, the UAVs coordinates are calculated using known GUs coordinates. Clusters of GU coordinates, the number of GUs and the optimal UAV positions Z are optimised based on the number of known GUs. As a result, the UAV coordinates are calculated using this algorithm. The altitude of the UAVs is then adjusted to match the heights of the obstacles. User identification, authentication, connection, and offloading are all part of the second stage. The connectivity of GUs and UAVs is limited by the max. number of allowable connections $\left(K_{b,max}\right)$ between the GU and the UAV. In this stage, $\left(K_{b,max}\right)$ is used as the input for each GU and UAV coordinate. A distance matrix $S(a, b)$ is generated for each UAV. Specific UAVs are assigned to specific GUs, and offloading occurs at the shortest possible distance between the UAV and the GU $(u_{ab} = 1)$. This means that if there are more connections than allowed, no link between the GU and UAV is established, and local execution takes place instead $(u_{ab} = 0)$. The Interior Point

Method is used in the third stage to transform non-linear constraints into linear ones. Following the establishment of the GU and UAV link, the interior point method can be used to find a linear solution to a given multi-objective problem function. Both linear and non-linear constraints can be used as inputs $\left(x0, A, A_{eq}, b_{eq}, lb, ub\right)$ are introduced. Following the above procedure, further three steps take place: Step I: s_i slack variables are derived from given non-linear constraints, which means non-linear constraints are converted into linear ones. Step II: Barrier function is introduced, which provides the approximation of a given objective function. Step III: If the approximated objective function is not locally convex and has not yet converged, the conjugate gradient method is then used to meet the convergence criteria. It is possible to maintain positive slack variables and thus adjust the radius and decision variables of the trusted region. This is accomplished by solving the approximated converged multi-objective resource allocation problem function using quadratic computations in conjunction with linear constraints.

5 Simulation Results

A UAV-MEC network is taken into consideration with $A = 20$ UAVs and $B = 150$ GUs. The network's bandwidth is set as $W = 1$ Mhz with channel power gain at 1 m reference distance as $d_o = 1.42e^{-4}$, and range as 500 m. The time block length set for all the tasks is $l = 6$ ms and min. data bits conveyed as $Q_b = 50$. The max. computation capacity available for the GU is set as $c_{b,max} = 1$ Ghz. The constant energy consumption of the GU during no offloading is set to be $C_b = 0.00001$W and max. available energy as $C_{max} = 20$ dBm. The max. GUs associated with the UAV is given as $K_{b,max} = 15$. The height interval for UAV is set as $H_a^{min} = 10, H_a^{max} = 20$ and elevation angle/half beamwidth interval for the antenna is $\theta_a^{min} = \pi/6, \theta_a^{max} = \pi/3$. The CPU model's positive constants are set as $\beta_b, v_b, w = 10^{-2e^9+1}, 1e^3, 2$.

Five different scenarios (1–5) were considered in this study, and the simulation results are depicted in Fig. 4. In each scenario, number of GUs are varied by deploying different number of UAVs for both no offloading and offloading conditions. Graph 1 shows the computation efficiency (utility function), graph 2 is for the number of computed bits, while energy consumption is depicted in graph 3.

As shown in Fig. 4, computation efficiency is composed of the number of computed bits by the energy consumed. In Scenario 1, by varying different GUs i.e., 25, 50, 75, 100, 125, 150, and deploying 4 UAVs for all those GUs are considered.For Scenarios 2, 25, 50, 75, 100, 125, 150 GUs are varied by deploying 8 UAVs. In case of Scenario 3, GUs i.e., 25, 50, 75, 100, 125, 150 are varied with 12 associated UAVs. 16 UAVs are linked with GUs 25, 50, 75, 100, 125, 150 in Scenario 4. In last Scenario 5, by different varying users 25, 50, 75, 100, 125, 150 by deploying 20 UAVs, results are calculated for both no offloading (local execution) and offloading cases.

Graph 1: Computation Efficiency Scenario 1 employs 4 UAVs by varying 25, 50, 75, 100, 125, 150 GUs. During local execution (no offloading) case, The computational efficiency of GUs is nearly constant when executed locally but significantly increases when the tasks are offloaded. In Scenario 2, where 8 UAVs are used for GUs 25, 50, 75, 100, 125, 150, offloading improves the computational efficiency of Scenario 2 slightly when compared to Scenario 1. Offloading gives better results as the number of GUs increases, which increases computation efficiency. Similarly, in the Scenario 3, 4, 5, the computation efficiency keeps on increasing during offloading but remains almost still during local execution.

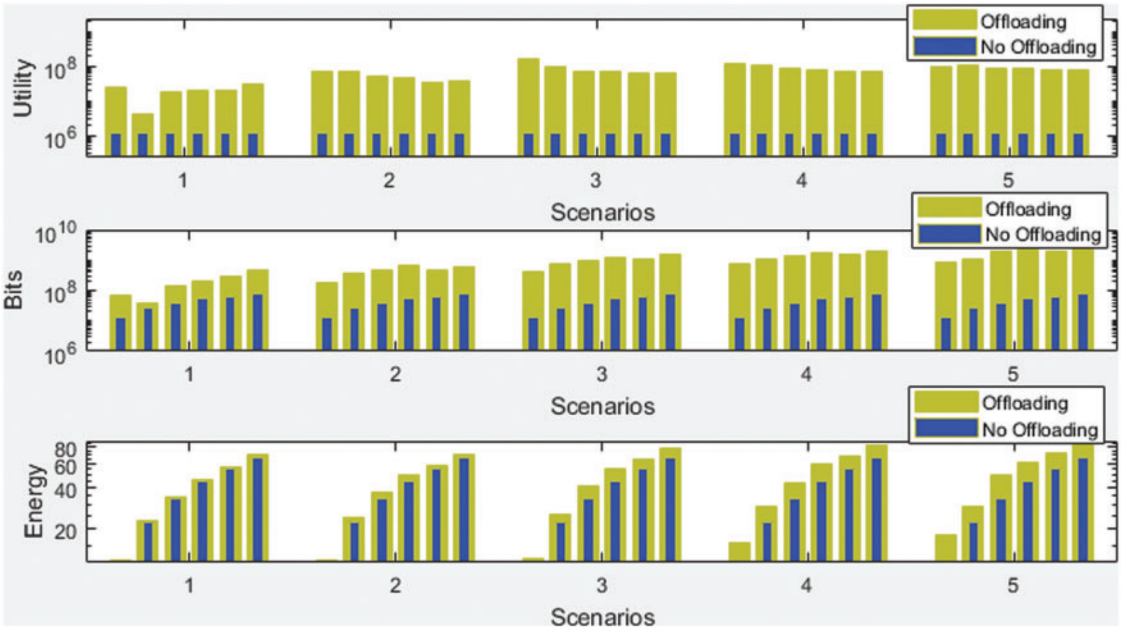

Figure 4: Simulation results

Graph 2: Number of Computed Bits Scenario 1 employs four UAVs while changing the no. of GUs, namely 25, 50, 75, 100, 125, 150. Scenarios 2, 3, 4 and 5 each employ eight, twelve, sixteen, and twenty UAVs.Each scenario shows a constant number of computed bits for local execution, but as the no. of GUs increases, the amount of offloading increases, as a result, number of computed bits increases for offloading case. Computed bits increases linearly with the number of GUs and UAVs.

Graph 3: Consumed Energy For Scenarios 1, 2, 3, 4, 5, MUs are varied i.e., 25, 50, 75, 100, 125, 150 by deploying 4, 8, 12, 16, 20 UAVs respectively. The third graph shows that when no offloading occurs, MUs consume less energy, which is not required for sure. But as the number of GUs and UAVs increases, the tasks are automatically start offloading to the UAVs. As the offloading increases, energy consumed during data transmission and reception between MU and UAV increases. However, it is shown in the graph that energy consumed during offloading is nearly equal to the local execution which means less energy is consumed during offloading, and it gives better results.

It is possible to assert that overall system performance has improved due to the increased number of computed bits and decreased energy consumption. In addition, offloading outperforms local execution, i.e., offloading improves computation efficiency more.

6 Conclusion

This paper investigates the computational efficiency for mobile IoT ground users by employing a UAV-MEC network with multiple rotary-winged UAVs and multiple users. In order to maximise the given objective function, which is constrained by latency, power, and quality of service requirements, we employ the K-means clustering algorithm and the interior point method, which are both implemented in MATLAB. Iterations are used to solve the problem that has been formulated. Specifically, small tasks (less data size) are completed locally by the ground user, whereas high computation tasks (large data size), are offloaded to the UAVs in order to improve the overall performance of the U-MEC system. Simulation Results show that offloading gives better results than the local execution. Non-Orthogonal Multiple Access (NOMA) is one technique that can further be used in future works.

Acknowledgement: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Evans, "The internet of things how the next evolution of the internet is changing everything," http://precog.iiitd.edu.in/people/anupama, 2011.
[2] Q. Pham and W. Hwang, "Resource allocation for heterogeneous traffic in complex communication networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 10, pp. 959–963, 2016.
[3] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*, Princeton university press, New Jersey, USA, 2012.
[4] R. I. Bor-Yaliniz, A. El-Keyi and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular net-works," in *2016 IEEE Int. Conf. on Communications (ICC)*, Kuala Lumpur, Malaysia, pp. 1–5, 2016.
[5] Y. Zeng, R. Zhang and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.
[6] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta *et al.,* "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-Edge Computing (MEC) Industry Initiative*, pp. 1089–7801, 2014.
[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, "Mobile-edge computing a key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.
[8] J. Xu, Y. Zeng and R. Zhang, "UAV-Enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5092–5106, 2018.
[9] Q. -V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le *et al.,* "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974117017, 2020.
[10] N. H. Motlagh, M. Bagaa and T. Taleb, "UAV-Based IoT platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.
[11] S. Jeong, O. Simeone and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2017.
[12] X. Zhang, Y. Zhong, P. Liu, F. Zhou and Y. Wang, "Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization," *IEEE Access*, vol. 7, pp. 113345–113354, 2019.
[13] J. Zhang, L. Zhou, F. Zhou, B. Seet, H. Zhang *et al.,* "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2114–2125, 2020.
[14] Y. Wang, Z. Ru, K. Wang and P. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2020.
[15] H. Guo and J. Liu, "UAV-Enhanced intelligent offloading for internet of things at the edge, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.

[16] Z. Yu, Y. Gong, S. Gong and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020.

[17] Y. Du, K. Wang, K. Yang and G. Zhang, "Energy-efficient resource allocation in UAV based MEC system for IoT devices," in *2018 IEEE Global Communications Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, pp. 1–6, 2018.

[18] T. Zhang, Y. Xu, J. Loo, D. Yang and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5505–5516, 2019.

[19] F. Zhou, Y. Wu, H. Sun and Z. Chu, "UAV-Enabled mobile edge computing: Offloading optimization and trajectory design," in *2018 IEEE Int. Conf. on Communications (ICC)*, Kansas City, USA, pp. 1–6, 2018.

[20] G. Wu, Y. Miao, Y. Zhang and A. Barnawi, "Energy efficient for UAV-enabled mobile edge computing networks: Intelligent task prediction and offloading," *Computer Communications*, vol. 150, pp. 556–562, 2020.

[21] Y. Liu, K. Xiong, Q. Ni, P. Fan and K. B. Letaief, "UAV-Assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2777–2790, 2020.

[22] P. Cao, Y. Liu, C. Yang, S. Xie and K. Xie, "MEC-Driven UAV- enabled routine inspection scheme in wind farm under wind influence," *IEEE Access*, vol. 7, pp. 179252–179265, 2019.

[23] B. Shang and L. Liu, "Mobile-edge computing in the sky: Energy optimization for air ground integrated networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7443–7456, 2020.

[24] M. Hua, Y. Wang, C. Li, Y. Huang and L. Yang, "UAV-Aided mobile edge computing systems with one by one access scheme," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 664–678, 2019.

[25] J. Hu, M. Jiang, Q. Zhang, Q. Li and J. Qin, "Joint optimization of UAV position, time slot allocation, and computation task partition in multiuser aerial mobile-edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7231–7235, 2019.

[26] X. Diao, J. Zheng, Y. Wu, Y. Cai and A. Anpalagan, "Joint trajectory design, task data, and computing resource allocations for NOMA-based and UAV-assisted mobile edge computing," *IEEE Access*, vol. 7, pp. 117448–117459, 2019.

[27] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao *et al.,* "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. vol. 10, pp. 187–10, 200, 2019.

[28] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao *et al.,* "Energy- efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.

[29] F. Zhou, Y. Wu, R. Q. Hu and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[30] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu *et al.,* "Energy aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles," *IEEE Access*, vol. 6, pp. 56700–56715, 2018.