Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: a digital twin approach

Received: 29 December 2021 | Revised: 19 February 2022 | Accepted: 28 February 2022

IET Signal Processing

DOI: 10.1049/sil2.12128

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: A digital twin approach

Yijiu Li[1] | Dang Van Huynh[1] | Tan Do-Duy[2] | Emi Garcia-Palacios[1] | Trung Q. Duong[1]

[1]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

[2]Department of Computer and Communications Engineering, HCMC University of Technology and Education, Hochiminh, Vietnam

Correspondence

Trung Q. Duong, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Queen's Road, Belfast BT7 1NN, UK.

Email: trung.q.duong@qub.ac.uk

Funding information

EPSRC-IAA, Grant/Award Number: 2020-2021; Engineering and Physical Sciences Research Council, Grant/Award Number: EP/P019374/1; Royal Academy of Engineering, Grant/Award Number: RCSRF2021\11\41

Abstract

A digital twin (DT) framework for Internet-of-thing (IoT) networks is proposed where unmanned aerial vehicles (UAVs) acting as flying mobile edge computing (MEC) servers support the task offloading on the fly. The considered DT model is very well suitable for industrial automation with the strict constraints of mission-critical services' ultra-reliable low-latency communication (URLLC) links. To support low-latency IoT devices, we formulate the end-to-end (e2e) latency minimisation problem of digital twin-aided offloading UAV-URLLC. Specifically, the minimised latency is obtained by jointly optimising both communication and computation parameters, namely power, offloading factors, and the processing rate of IoT devices and MEC-UAV servers. Due to the highly non-convex optimisation problem, we first consider the K-means clustering algorithm to optimally deploy the on-demand UAVs. Then, an alternative optimisation approach combined with appropriate inner approximations is effectively exploited to tackle this challenge. We demonstrate the effectiveness of the proposed DT framework through representative numerical results.

KEYWORDS

alternating optimisation, digital twin, edge networks, UAV, ultra-reliable and low latency communications (URLLC)

1 | INTRODUCTION

1.1 | Literature review

Digital Twin (DT) is an emerging technology that is able to create virtual twins of physical objects in order to facilitate the processing of control and to manage cyber-physical systems. DT can be exploited in networking and communications for many aspects, such as system modelling, physical data processing, cloud computing, and edge computing [2]. Therefore, studies of DT are attracting much attention from active researchers [3–6]. More specifically, in [3], a DT-assisted task offloading in mobile edge computing (MEC) was investigated to address the problem of minimising power and time overhead. Another work in reducing offloading latency for DT edge network was introduced in [4]. In [5], DT was proposed for intelligent authorisation in the beyond 5G smart grid applications. DT was exploited in [6, 7] to empower edge networks for the industrial Internet of things environment. These representative studies demonstrate the huge potential of DT in various domains, especially in networked systems.

In recent years, unmanned aerial vehicles (UAVs) have been under the spotlight due to their flexible configuration and mobile characteristics [8, 9]. Numerous research studies have been conducted to enhance the control performance of UAVs. In [10], authors carried out the ground test of UAV, evaluating the performance of its entire flight control system through rotor speed, roll attitude, etc. The flight test [11] and collision

This paper has been accepted in part for presentation in the first International Conference on 6G Networking (6GNet 2022) to be held on July 06-08, 2022 in Paris, France [1].

avoidance [12] have also been studied to give a stronger control over this smart vehicle. UAVs achieve even better performance in many research areas by combining with other advanced technologies. To integrate with the intelligent reflecting surface (IRS) [13], a well-performed UAV-assisted IRS symbiotic radio system has been formed. The system performs better as data information is transferred via UAV by optimising the UAV trajectory and the IRS phase shifts. Applying a deep reinforcement learning algorithm [14], the decision making of UAVs can be autonomous rather than pre-planned. UAV's low consumption of energy also attracts public attention. Through optimisation algorithms, resource allocation can be optimised, and the total energy consumption in a multi-UAV network framework can be minimised [15, 16]. Based on their outstanding advantages, UAVs are now being developed and used in military and civil applications [17]. These applications are in many fields, such as environment monitoring, traffic control, public safety, damaged buildings detection, and industrial automation [18–20]. Particularly, with the help of UAV, Yang X et al. [21] develop a method for high-precision geolocation of distant targets, which is more effective than the conventional one-shot localisation way. Search and rescue operations with UAVs participation increase the speed of rescue and thus improve the survival rate of people [22, 23]. In Ref. [24], UAVs are used as flying base stations to ensure the connectivity of communication networks in unexpected disasters. These aerial vehicles also play a role in smart cities [19].

With the rapid development of the 5G network, ultra-reliable and low-latency communication (URLLC) emerges as a promising paradigm to ensure a certain quality of service (QoS). With strict requirements of extremely low latency (from 1 ms to few milliseconds) and ultra-high reliability (over 99.999%) [25], URLLC plays an indispensable role in remote healthcare, autonomous driving, immersive virtual reality, cloud robotics, deterministic communication, and many other areas [26]. This novel communication service uses short packet transport, which allows optimising the transmission of control information [27].

Evolved from cloud computing, MEC has been widely considered as a key application in 5G communication. This promising technology extends the capabilities of cloud computing at the network edge [28] and performs excellently in smart manufacturing, industrial Internet of things (IoT) as well as many other areas [29–31]. In recent years, many efforts have been put into MEC. In Ref. [32], the author demonstrates a well-established MEC architecture and integrates an application deployment use case, establishing a proof of concept, which is very similar to the actual deployment of the MEC system in a 5G environment. Combining with the optimising method, MEC is an appropriate solution to improve the quality of service. In Ref. [33], the author proposes a Reinforcement Learning (RL)-based optimisation framework to minimise the cost of delay and energy consumptions for user equipment in a time-variant dynamic MEC system. The considered MEC system outperforms other baseline solutions according to the demonstration, whereas Wu J et al. [34] adopt an offloading strategy in MEC that considers delay and energy consumptions of cost optimisation. Two schemes named optimised OMA and hybrid NOMA are proposed in [35] for solving the problem of joint power and time allocation for MEC offloading. Through these extensive studies of MEC, this evolving technology has been used in various use cases, for example, an audience metre [36]. In this particular use case, MEC modules are used to improve the algorithm's performance, detecting the estimated number of participants in an event over the entire time period.

MEC and URLLC techniques are closely related to each other. Under the sufficiently powerful computing of MEC, applications can be processed in real time. By reducing the transmission processing time and reception processing time, MEC can reduce latency in 5G systems significantly. Since finite blocklength (FBL) is adopted to satisfy the latency constraints of URLLC, Yang et al. [37] propose a MEC network where one MEC server is used to minimise the error probability between users under FBL and energy consumption constraints. This FBL scheme is also considered in a MEC-enabled vehicular network to support URLLC [38]. The placement of MEC servers that promises URLLC requirements has also been studied. An algorithm called LowMEP has been proposed to find a minimum number of MEC servers, satisfying the quality of service [39]. The combination of these two promising techniques is frequently used in various fields. For industrial application, Jia et al. [40] propose a 5G MEC gateway system capable of supporting URLLC to enable communication in factories. For autonomous vehicles application, vehicle's latency and energy cost functions have been established to investigate URLLC resource scheduling for edge computing [41].

Additionally, UAV-assisted communications can be effectively combined with MEC to enable time-sensitive and computation-intensive services for a wide range of Internet of Things (IoT) applications [42–45]. UAV-based edge networks not only be able to minimise the energy consumption of IoT devices with optimal offloading decisions [46, 47] but also reduce the latency by providing edge caching solutions [48]. More importantly, ultra-reliable and low-latency communications (URLLC) recently emerged as a promising technology for mission-critical applications [49]. Combining UAV-enabled MEC with URLLC opens many opportunities as well as challenges for the next generation of IoT applications [50].

1.2 | Motivations and main contributions

Recently, combined merging technologies, including MEC, DT, UAV, and URLLC, are attracting many active research groups [3, 4, 42]. In particular, the DT-assisted task offloading based on edge collaboration has been investigated in [3] with a DRL-based approach. The edge selection and offloading optimisation have been addressed in this paper; however, the communication resource optimisation has not taken into consideration. Similarly, in [4, 42], the joint computation and communication resources have not been fully addressed to obtain the optimal latency of task offloading. More

importantly, the combination of DT, MEC with URLLC for industrial scenarios is a promising research direction, which still has many open issues that can be further explored and contributed to this research area.

Moving beyond above background, this paper addresses the problem of combining these important technologies, namely DT, MEC, and URLLC, in UAV-assisted IoT systems. Both communication and computation factors, including transmit power, the processing rate of IoT devices or user equipment (UE), edge servers, and offloading policies, are carefully taken into consideration to reduce the end-to-end (e2e) latency. Main contributions of this paper are summarised as follows:

- We first formulate the latency minimisation problem of UAV-based edge network with URLLC in the DT regime. The addressed problem fully considers both communication and computation factors in reducing the task offloading latency.
- In order to solve the problem, we propose the AO-IA algorithm with two sub-problems, namely transmit power and computation resources optimisation, offloading portions of optimisation.
- Finally, intensive simulations have been conducted to demonstrate the effectiveness of the proposed solution.

The rest of the article is structured as follows: Section 2 presents the system model and problem formulation for attaining minimum e2e latency. Section 3 resolves the problem formed in the previous section using an algorithmic solution. Numerical results are discussed in Section 4. Finally, Section 5 concludes the article.

2 | SYSTEM MODEL AND PROBLEM FORMULATION

This section leads towards the problem to be solved for minimising the e2e latency by expressing the basic network model, transmission model, DT-empowered offloading, associated energy and power consumption model, and the UAV deployment.

2.1 | DT-empowering URLLC-based edge networks model

Figure 1 presents a DT-enabled UAV-based edge network architecture with URLLC. The physical layer consists of IoT devices (IoT), also known as UEs, and UAVs. These physical devices connect via URLLC links to ensure stringent reliability and low-latency communications in mission-critical applications.

Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be the set of M IoT devices and $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of K UAVs. There are K UAV-IoT groups, in which the kth UAV serves M_k IoT devices in each group. Each UAV can act as an access point (AP) with the capability to perform as an edge server (ES). We assume that the UAV deployment and network planning are performed in advance.

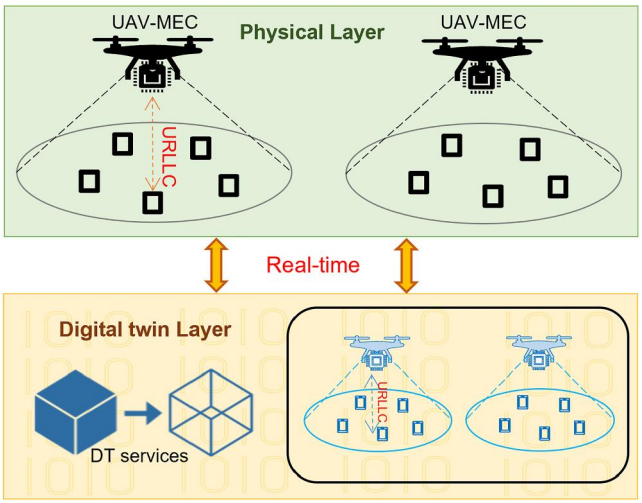

FIGURE 1 An exemplary illustration of the DT-enabled UAV-based edge networks with ultra-reliable low-latency communications (URLLC) [1]

2.2 | Transmission model

2.2.1 | Channel model

The air-to-ground (ATG) channels between UAVs and UEs are also dominant by light-of-sight (LoS) propagation but these are more complex due to the effects of propagation attenuation by blockage geometry and shadowing [51]. As such, the path loss of the link between the kth UAV and the (m, k)-th UE can be written as

$$g_{mk} = PL_{mk} + \eta^{LoS} P^{LoS}_{m,k} + \eta^{NLoS} P^{NLoS}_{mk}, \tag{1}$$

where η^{LoS} and η^{NLoS} are the average additional losses for LoS and non LoS (NLoS), respectively. The path loss in respect to the distance (PL_{mk}) is given by

$$PL_{mk} = 10 \log\left(\frac{4\pi f_c r_{mk}}{c}\right)^{\beta}, \tag{2}$$

where f_c is carrier frequency (Hz), c is the speed of light (m/s), $\beta \geq 2$ is the path loss exponent, $r_{mk} = \sqrt{d^2_{mk} + Z^2_k}$, d_{mk} is the Euclidean distance between the mth UE and the kth UAV, and Z_k is the antenna height of the kth UAV. The probability of LoS and NLoS can be shown as

$$P^{LoS}_{mk} = \frac{1}{1 + a \exp\left[-b\left(\arctan\left(\frac{Z_k}{d_{mk}}\right) - a\right)\right]} \tag{3}$$

$$P^{NLoS}_{mk} = 1 - P^{LoS}_{mk}, \tag{4}$$

where the constants a and b depend on the specific arrangement of the environment.

Each UAV is equipped with L antennas to serve M_k single-antenna IoT devices. Let $\mathbf{h}_{mk} \in \mathbb{C}^{L\times 1}$ be the channel vector between the kth UAV and the mth IoT, which can be modelled as $\mathbf{h}_{mk} = \sqrt{g_{mk}}\overline{\mathbf{h}}_{mk}$. Here, g_{mk} denotes the large-scale channel coefficient defined in (1), and $\overline{\mathbf{h}}_{mk}$ is the small-scale fading following the distribution of $\mathcal{CN}(0,\mathbf{I})$. Let $\mathbf{H}_k \in \mathbb{C}^{L\times M_k}$ be the channel matrix from M_k devices to the kth UAV with $\mathbf{H}_k = \left[\mathbf{h}_{k1}, \mathbf{h}_{k2}, .., \mathbf{h}_{kM_k}\right]$. Under the shared wireless medium, the $L \times 1$ received signal vector at the kth UAV is given by $\mathbf{y}_k = \sum_{m=1}^{M_k}\mathbf{h}_{mk}\sqrt{p_{mk}}s_{mk} + \mathbf{n}_k$, where p_{mk} is the payload power of the (m, k)-th device, s_{mk} is the zero mean and unit variance Gaussian information message from the (m, k)-th IoT, and $\mathbf{n}_k \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_L)$ is the additive white Gaussian noise (AWGN) during the data transmission with N_0 being the noise power.

In this paper, we consider the uplink transmission from IoTs to UAVs to perform task offloading. We apply the maximum-ratio combining (MRC) at the UAV to improve the performance gain. Moreover, to guarantee fairness among all IoT devices and further improve wireless transmission performance, we additionally adopt the matched filter and successive interference cancellation (MF-SIC) technique at the UAVs. In particular, by using MF-SIC, we assume that the decoding order follows IoTs' index by arranging the channel vector as $\|h_{1k}\|^2 \geq \|h_{2k}\|^2 \ldots \geq \|h_{M_k k}\|^2, \forall k$. Consequently, the signal-to-interference-plus-noise (SINR) at the kth UAV of the signal from the (m, k)-th IoT device can be expressed as

$$\gamma_{mk}(\mathbf{p}) = \frac{p_{mk}\|\mathbf{h}_{mk}\|^2}{\mathcal{I}_{mk}(\mathbf{p}) + N_0}, \tag{5}$$

where $\mathcal{I}_{mk}(\mathbf{p}) = \sum_{n>m}^{M} p_{nk}\|\mathbf{h}_{nk}\|^2$ is the interference power caused by IoT devices $n > m$ and $\mathbf{p} = \{p_m\}_{\forall m}$

2.2.2 | URLLC-based uplink transmission rate

The approximation of achievable transmission rate (bit/s) in URLLC finite blocklength is [52, 53]:

$$R_{mk}(\mathbf{p}) \approx (1-\omega_k)B\log_2[1+\gamma_{mk}(\mathbf{p})] - B\sqrt{\frac{(1-\omega_k)V_{mk}(\mathbf{p})}{N}}\frac{Q^{-1}(\epsilon)}{\ln 2}, \tag{6}$$

where $\omega_k = M_k/N$, $\forall k$, N is the blocklength, which can be written as $N = \delta B$ with B as the bandwidth and δ as the transmission time interval; ϵ is the decoding error probability, $\gamma_{mk}(\mathbf{p})$ denotes the SINR, $Q^{-1}(.)$ is the inverse function $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(\frac{-t^2}{2}\right)dt$, and V is the channel dispersion given by $V_{mk}(\mathbf{p}) = 1 - [1+\gamma_{mk}(\mathbf{p})]^{-2}$. When the blocklength N approaches to infinity, the data rate R_{mk} approaches $(1-\omega_k)B\log_2(1+\gamma_{mk}(\mathbf{p}))$, which is the classic Shannon's equation.

2.3 | Digital twin empowered task offloading model

A particular task from the (m, k)-th IoT device is represented by a tuple $J_{mk} = \{D_{mk}, C_{mk}, T_{mk}\}$, where D_{mk} is data size (bits), C_{mk} is required computation resource (cycles), and T_{mk} (s) is the minimum required latency for task J_{mk}.

Let $\boldsymbol{\alpha} = \{\alpha_{mk}\}_{\forall m,k}$ be the amount of the task processed locally, which satisfies $0 \leq \alpha_m \leq 1$.

2.3.1 | Local processing

For the (m, k)-th IoT device, its DT (DT_{mk}) can be expressed as

$$\mathrm{DT}_{mk} = \left(f_{mk}^{\mathrm{loc}}, \hat{f}_{mk}^{\mathrm{loc}}\right), \tag{7}$$

where f_{mk}^{loc} is the estimated processing rate of the physical IoT device, and $\hat{f}_{mk}^{\mathrm{loc}}$ is the deviation between the real device and its DT.

The DT layer has the estimated processing rate f_{mk}^{loc} to replicate the behaviours of IoT devices and trigger decisions on optimising physical devices configuration.

The (m, k)-th IoT executes α_{mk} portion of task J_{mk} with the estimated processing rate f_{mk}^{loc}, and the estimated time required to execute the task locally is given by

$$\tilde{T}_{mk}^{\mathrm{loc}}\left(\alpha_{mk}, f_{mk}^{\mathrm{loc}}\right) = \frac{\alpha_{mk}C_{mk}}{f_{mk}^{\mathrm{loc}}}. \tag{8}$$

Assuming that the deviation between the physical IoT ($\mathcal{M}$) and $\tilde{\mathcal{M}}$ in DT can be acquired in advance, the computing latency gap between real value and DT estimation is computed as

$$\Delta T_{mk}^{\mathrm{loc}}\left(\alpha_{mk}, f_{mk}^{\mathrm{loc}}\right) = \frac{\alpha_{mk}C_{mk}\hat{f}_{mk}^{\mathrm{loc}}}{f_{mk}^{\mathrm{loc}}\left(f_{mk}^{\mathrm{loc}} - \hat{f}_{mk}^{\mathrm{loc}}\right)}. \tag{9}$$

The actual time for local computing is expressed as

$$T_{mk}^{\mathrm{loc}} = \Delta T_{mk}^{\mathrm{loc}} + \tilde{T}_{mk}^{\mathrm{loc}}. \tag{10}$$

2.3.2 | Edge processing

Given the estimated processing rate of the kth ES for executing the offloaded task from the (m, k)-th IoT device is f_{mk}^{es}, the estimated latency of the kth ES to execute task J_m is given by

$$\tilde{T}_{mk}^{\mathrm{es}}(\alpha_{mk}, f_k^{\mathrm{es}}) = \frac{(1-\alpha_{mk})C_{mk}}{f_{mk}^{\mathrm{es}}}. \tag{11}$$

Then, the latency gap $\Delta T_{mk}^{\mathrm{es}}$ between the real value and DT estimation can be expressed as

$$\Delta T_{mk}^{\mathrm{es}}\left(\alpha_{mk}, f_{k}^{\mathrm{es}}\right)=\frac{(1-\alpha_{mk}) C_{m} \hat{f}_{mk}^{\mathrm{es}}}{f_{mk}^{\mathrm{es}}\left(f_{mk}^{\mathrm{es}}-\hat{f}_{mk}^{\mathrm{es}}\right)}. \tag{12}$$

As a result, the actual latency for executing at edge DT can be expressed as

$$T_{mk}^{\mathrm{es}}=\Delta T_{mk}^{\mathrm{es}}+\tilde{T}_{mk}^{\mathrm{es}}. \tag{13}$$

The total DT latency in the system can be expressed as follows:

$$\begin{aligned} T_{mk}^{\mathrm{tot}} &= T_{mk}^{\mathrm{loc}}+T_{mk}^{\mathrm{comm}}+T_{mk}^{\mathrm{es}} \\ &= \frac{\alpha_{mk} C_{m}}{f_{mk}^{\mathrm{loc}}-\hat{f}_{mk}^{\mathrm{loc}}}+\frac{(1-\alpha_{mk}) D_{mk}}{R_{mk}(\mathbf{p})}+\frac{(1-\alpha_{mk}) C_{mk}}{f_{mk}^{\mathrm{es}}-\hat{f}_{mk}^{\mathrm{es}}}. \end{aligned} \tag{14}$$

The latency comprises three main components, namely local processing latency $\left(T_{mk}^{\mathrm{loc}}\right)$, uplink transmission latency $\left(T_{mk}^{\mathrm{comm}}\right)$, and edge processing latency $\left(T_{mk}^{\mathrm{es}}\right)$. Since the response messages from UAVs to IoTs are typically small (e.g. control packets), the downlink transmission latency is negligible [54, 55].

2.4 | Energy and power consumption model

Total energy consumption of the (m, k)-th IoT includes energy for transmission and computation:

$$\begin{aligned} E_{mk}^{\mathrm{tot}}(\alpha_{mk}, \boldsymbol{\beta}, \mathbf{p}) &= E_{mk}^{\mathrm{comp}}+E_{mk}^{\mathrm{comm}}=\frac{\theta_{m}}{2} \alpha_{mk} C_{mk}\left(f_{mk}^{\mathrm{loc}}-\hat{f}_{mk}^{\mathrm{loc}}\right)^{2} \\ &\quad+\frac{(1-\alpha_{mk}) p_{mk} D_{mk}}{R_{mk}(\mathbf{p})}, \end{aligned} \tag{15}$$

where $\theta_m/2$ represents the average switched capacitance and the average activity factor of the mth IoT [56].

The power consumption of the kth UAV for processing the uploaded tasks is modelled as follows [48]:

$$P_{k}\left(\alpha_{mk}, f_{mk}^{\mathrm{es}}\right)=\sum_{m \in \mathcal{M}_{k}}(1-\alpha_{mk})\left(f_{mk}^{\mathrm{es}}\right)^{3} \theta_{k}, \tag{16}$$

where θ_k represents the average switched capacitance and the average activity factor of the kth UAV.

2.5 | UAV deployment

In this section, we present the clustering algorithm for UAV deployment by considering an efficient QoS-constrained K-means clustering approach [57–59]. In particular, the constrained clustering method considers whether the mth UE can be grouped in the kth cluster based on two types of pairwise constraints, namely must-link constraints and cannot-link constraints [58], which represent the satisfied QoS constraints and the violated QoS constraints, respectively.

The constrained K-means clustering algorithm can be briefly described as follows. At the initial stage, the locations of UAVs are randomly set within the deployment area as the centroid location. First, based on the Euclidean distance between the UEs and the UAVs, the mth UE is assigned to an appropriate cluster with the smallest distance. Then, if QoS constraints are not satisfied for any UEs, the altitude of the corresponding UAV must be adjusted. Finally, the centroid location for each cluster is updated. Such procedure repeats until the cluster members are stable or the number of iterations exceeds a predefined threshold.

As an illustrative case, we set the number of UEs randomly located in a critical area at $M = 6$ and the number of UAVs at $K = 2$ with the path loss threshold corresponding to the QoS requirement $\gamma_{QoS} = 110$ dB. Figure 2 shows the clustering result after implementing the QoS-constrained K-means clustering algorithm.

2.6 | Problem formulation

Here, the worst case of the total DT latency is minimised by optimising offloading policies, transmit power, and estimated processing rates of IoT and ESs. By defining the following notations $\mathcal{D} \triangleq \{\alpha_{mk}, \forall m, k | 0 \leq \alpha_{mk} \leq 1, \forall m, k\}$, $\mathcal{P} \triangleq \{p_{mk}, \forall m, k | 0 \leq p_{mk} \leq P_{mk}^{\max}, \forall m\}$, and $\mathcal{F} \triangleq \left\{f_{mk}^{\mathrm{loc}}, f_{mk}^{\mathrm{es}}, \forall m, k | 0 \leq f_{mk}^{\mathrm{loc}} \leq F_{\max}^{\mathrm{loc}}, \forall m; 0 \leq f_{mk}^{\mathrm{es}} \leq F_{\max}^{\mathrm{es}}, \forall k\right\}$ as the set constraints of offloading decisions, uplink transmission power, processing rates, respectively, the problem is formulated as follows:

$$\min_{\boldsymbol{\alpha}, \mathbf{p}, \mathbf{f}} \max_{\forall m \in \mathcal{M}} \left\{T_{m}^{\mathrm{tot}}(\alpha_{mk}, \mathbf{f}, \mathbf{p})\right\}, \tag{17a}$$

$$\text{s.t. } T_{mk}^{\mathrm{tot}}(\alpha_{mk}, \mathbf{f}, \mathbf{p}) \leq T_{mk}^{\max}, \forall mk, \tag{17b}$$

$$R_{mk}(\mathbf{p}) \geq R_{\min}, \forall m, k, \tag{17c}$$

$$E_{mk}^{\mathrm{tot}}(\alpha_{mk}, \mathbf{p}) \leq E_{\max}, \forall m, \tag{17d}$$

$$\sum_{m \in \mathcal{M}_{k}}(1-\alpha_{mk}) f_{mk}^{\mathrm{es}} \leq F_{\max}^{\mathrm{es}}, \forall k, \tag{17e}$$

$$P_{k}\left(\alpha_{mk}, f_{k}^{\mathrm{es}}\right) \leq P_{\max}^{\mathrm{es}}, \forall k, \tag{17f}$$

$$\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{D}, \mathbf{p} \in \mathcal{P}, \mathbf{f} \in \mathcal{F}, \tag{17g}$$

where constraint (17b) presents maximum latency constraint for every incoming task. Constraints (17c) and (17d) are the minimum transmission rate requirements for uplink transmission and the maximum energy consumption requirement of IoT, respectively. Finally, constraints (17e) and (17f) refer to maximum available computation resource and power budget of the UAVs.

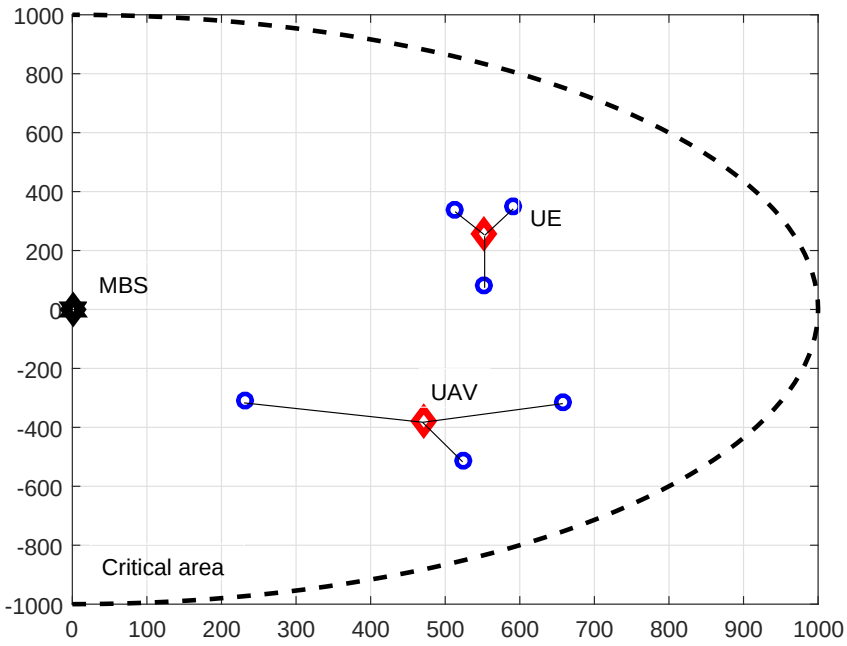

FIGURE 2 An illustrative system model with $M = 6$ UEs and $K = 2$ unmanned aerial vehicles (UAVs) after the implementation of QoS-constrained K-means clustering algorithm

3 | PROPOSED SOLUTIONS

As we can observe from (17), the objective function is non-concave and non-smooth, while other constraints (17c), (17d), (17e), and (17f) are also highly complex non-convex constraints. This results in solving the problem directly, which is computationally challenging. Therefore, to solve the problem (17), we first replace the objective function by an upper bound function with introduced variables $\mathbf{t} \triangleq \{t_{lc}, t_{cm}, t_{ed}\}$ satisfying $\tau_m(t_{lc}, t_{cm}, t_{ed}) \triangleq t_{lc} + t_{cm} + t_{ed}$ to equivalently transforms (17) to

$$\min_{\boldsymbol{\alpha},\mathbf{p},\mathbf{f},\mathbf{t}} \quad \max_{\forall m,k}\{\tau_{mk}(\mathbf{t})\}, \tag{18a}$$

$$\text{s.t.} \quad (17c) - (17g) \tag{18b}$$

$$\tau_m(\mathbf{t}) \le T_m^{\max}, \forall m, \tag{18c}$$

$$t_{lc} \ge \frac{\alpha_{mk} C_{mk}}{f_{mk}^{\text{loc}} - \hat{f}_{mk}^{\text{loc}}}, \forall m, k \tag{18d}$$

$$t_{cm} \ge \frac{(1-\alpha_{mk}) D_{mk}}{R_{mk}(\mathbf{p})}, \forall m, k \tag{18e}$$

$$t_{ed} \ge \frac{(1-\alpha_{mk}) C_{mk}}{f_{mk}^{\text{es}} - \hat{f}_{mk}^{\text{es}}}, \forall m, k, \tag{18f}$$

Lemma 1 *If* $(\boldsymbol{\alpha}^\star, \boldsymbol{p}^\star, \boldsymbol{f}^\star, \boldsymbol{t}^\star)$ *is the optimal solution to problem (18), then* $(\boldsymbol{\alpha}^\star, \boldsymbol{p}^\star, \boldsymbol{f}^\star, \boldsymbol{t}^\star)$ *is also the optimal solution to problem (17) and vice versa.*

Proof To prove Lemma 1, we show that the constraints (18d)–(18f) must hold with equality at optimum. Firstly, assuming that the equality of (18e) does not hold at the optimum for some m, that is, existing $t_{lc} > \frac{\alpha_{mk}^\star C_{mk}}{f_m^{\text{loc}\star} - \hat{f}_m^{\text{loc}}}$. There exists a positive constant $\Delta t_{\text{lc}} > 0$, which is defined as $t_{lc} - \Delta t_{lc} = \frac{\alpha_{mk}^\star C_{mk}}{f_m^{\text{loc}\star} - \hat{f}_m^{\text{loc}}}$. As a result, $t_{lc} - \Delta t^{\text{loc}}$ is also feasible to problem (18), but results in a strictly lower latency. This contradicts the original assumption that the set $(\boldsymbol{\alpha}^\star, \mathbf{p}^\star, \mathbf{f}^\star, \mathbf{t})$ is the optimal solution to problem (18). Other constraints (18e), (17f) are followed similarly. □

Due to the complexity of the non-convex problem (18), we decompose (18) into two sub-problems and solve the problem in the fashion of alternating optimisation (AO) approach and inner approximation (AO-IA) framework [60, 61]. The following subsections fully present the development of our proposed solution.

3.1 | Transmit power and computation resource optimisation

In this subsection, we solve (18) for given $(\alpha^{(i)})$ to obtain the next optimal values of $(\mathbf{p}^{(i+1)}, \mathbf{f}^{(i+1)})$

$$\underset{\mathbf{p},\mathbf{f},\mathbf{t}|\boldsymbol{\alpha}^{(i+1)}}{\text{minimize}} \quad \max_{\forall m,k}\{\tau_{mk}(\mathbf{t})\}, \tag{19a}$$

$$\text{s.t.} \quad (17c), (17d), (17e), (17f), (17g), (18c), (18d), (18e), (18f). \tag{19b}$$

As we can observe from the sub-problem (19), the constraints (17c), (17d), and (18e) are non-convex. We are now in the position to approximate these constraints.

Convexify of (17c): To address constraint (17c), we first rewrite that $\gamma_{mk}(\mathbf{p}) = \frac{p_{mk}}{q_{mk}(\mathbf{p})}$, where $q_{mk}(\mathbf{p})$ is defined as

$$q_{mk}(\mathbf{p}) \triangleq \frac{\mathcal{I}_{mk}(\mathbf{p}) + N_0}{\|\mathbf{h}_{mk}\|^2} \tag{20}$$

Following the Appendix, we have

$$R_{mk}(\mathbf{p}) \geq R_{mk}^{(i)}(\mathbf{p}) \triangleq \frac{(1-\omega_k)B}{\ln 2}\left[\mathcal{G}_{mk}^{(i)}(\mathbf{p}) - \kappa \mathcal{W}_{mk}^{(i)}(\mathbf{p})\right] \tag{21}$$

under the trusted regions.

$$q_m(\mathbf{p}) + p_{mk} \leq 2\left(q_m\left(\mathbf{p}^{(i)}\right) + p_{mk}^{(i)}\right), \ \forall m,k, \tag{22}$$

$$\frac{q_m(\mathbf{p}) + p_{mk}}{q_m(\mathbf{p}^{(i)}) + p_{mk}^{(i)}} \leq 2\frac{q_m(\mathbf{p})}{q_m(\mathbf{p}^{(i)})}, \ \forall m,k, \tag{23}$$

where $\mathcal{G}_{mk}^{(i)}(\mathbf{p})$, and $\mathcal{W}_{mk}^{(i)}(\mathbf{p})$ are defined as in the Appendix, $\kappa = \frac{Q^{-1}(\epsilon)}{\sqrt{(1-\omega)N}}$. As a results, we innerly approximate constraint (17e) as

$$R_{mk}^{(i)}(\mathbf{p}) \geq R_{\min}, \forall m,k. \tag{24}$$

Convexify of (17d): By introducing variables $\mathbf{r} \triangleq \{r_{mk}\}_{\forall m,k}$ that satisfy $r_{mk} \geq 1/R_{mk}$, $\forall m, k$. We can equivalently express (17d) as follows

$$\begin{cases} \frac{\theta}{2}\alpha_{mk}^{(i+1)} C_{mk}\left(f_{mk}^{\text{loc}} - \hat{f}_{mk}^{\text{loc}}\right)^2 + (1-\alpha_{mk}^{(i+1)}) D_{mk} p_{mk} r_{mk} \leq E_m^{\max}, \ \forall m,k, & \text{(25a)} \\ \frac{1}{R_{mk}^{(i)}(\mathbf{p})} \leq r_{mk}. \ \forall m,k & \text{(25b)} \end{cases}$$

The constraint (25b) is now convex, while (25a) is still non-convex so we apply the following inequality

$$xy \leq \frac{1}{2}\left(\frac{\overline{y}}{\overline{x}}x^2 + \frac{\overline{x}}{\overline{y}}y^2\right), \tag{26}$$

with $x = p_{mk}, \overline{x} = p_{mk}^{(i)}, y = r_{mk}, \overline{y} = r_{mk}^{(i)}$ to approximate (25a) as

$$\begin{aligned} &\frac{\theta}{2}\alpha_{mk}^{(i+1)} C_{mk}\left(f_{mk}^{\text{loc}} - \hat{f}_{mk}^{\text{loc}}\right)^2 + \left(1-\alpha_{mk}^{(i+1)}\right) \\ &\quad \times D_{mk}\frac{1}{2}\left(\frac{r_{mk}^{(i)}}{p_{mk}^{(i)}}p_{mk}^2 + \frac{p_{mk}^{(i)}}{r_{mk}^{(i)}}r_{mk}^2\right) \leq E_m^{\max}, \ \forall m,k, \end{aligned} \tag{27}$$

which is now a convex constraint.

Convexify of (18e): By using $\mathbf{r}$ defined in Equation (25b), Equation (18e) can be convexified as

$$t_{cm} \geq \left(1-\alpha_{mk}^{(i+1)}\right) D_{mk} r_{mk}, \ \forall m,k, \tag{28}$$

Based on the above developments, we solve the following approximate convex program of (19) at iteration ith:

$$\underset{\mathbf{p},\mathbf{f},\mathbf{t}|\boldsymbol{\alpha}^{(i+1)}}{\text{minimize}} \quad \max_{\forall m,k}\{\tau_{mk}(\mathbf{t})\}, \tag{29a}$$

$$\text{s.t.} \quad (17e), (17f), (17g), (18c), (18d), (18f), (22), (23), (24), (25b), (27), (28). \tag{29b}$$

This is a convex program so that we can solve it efficiently with the CVX package. For complexity analysis, the convex problem (29) comprises 11 $KM_k + 2K$ linear or quadratic constraints and $5KM_k + K$ scalar decision variables, which leads to the per-iteration computational complexity of $\mathcal{O}\left(\sqrt{11\ KM_k + 2K}\ (5KM_k + K)\right)$[57, 62].

Algorithm 1 AO-IA-based Algorithm for solving (18)

1: **Input:** Set $i = 0$ and randomly choose initial feasible points $\mathcal{S}_1^{(0)}$ and $\mathcal{S}_2^{(0)}$ to constraints in (30), (29)
Set the tolerance $\epsilon = 10^{-3}$ and the maximum number of iterations $I_{\max} = 20$.
2: **Repeat**
3: Solve problem (29) with given $\mathcal{S}_2^{(i)}$ to obtain the optimal solution of $(\mathbf{p}^*, \mathbf{f}^*, \mathbf{r}^*)$ and update $(\mathcal{S}_1^{(i+1)} := (\mathbf{p}^*, \mathbf{f}^*, \mathbf{r}^*)$;
4: Solve problem (30) with given $\mathcal{S}_1^{(i+1)}$ to obtain the optimal solution of (α^*) and update $\mathcal{S}_2^{(i+1)} := (\alpha^*)$;
5: Set $i := i + 1$;
6: **Until** Convergence or $i > I^{\max}$.
7: **Output:** $\{\alpha^*, \beta^*, \mathbf{p}^*, \mathbf{f}^*\}$ and $\max\{\tau_{mk}(\mathbf{t})\}_{\forall m,k}$.

3.2 | Task offloading optimisation

In this subsection, we solve (18) for given $(\mathbf{p}^{(i)}, \mathbf{f}^{(i)})$ to obtain the next optimal values of offloading policies $(\boldsymbol{\alpha}^{(i+1)})$. The observed optimisation is now given by

$$\underset{\boldsymbol{\alpha},\mathbf{t}|\mathbf{p}^{(i)},\mathbf{f}^{(i)}}{\text{minimize}} \quad \max_{\forall m,k}\{\tau_{mk}(\mathbf{t})\}, \tag{30a}$$

$$\text{s.t.} \quad (17d), (17e), (17f), (17g), (18c), (18d), (18e), (18f). \tag{30b}$$

This is obviously a convex program and can be solved effectively with standard solvers, such as CVX [63]. The per-iteration of solving this convex program is $\mathcal{O}\left(\sqrt{7KM_k + 2K}\ .4KM_k\right)$.

3.3 | Proposed algorithm

Let us denote $\mathcal{S}_1^{(i)} \triangleq \left(\mathbf{p}^{(i)}, \mathbf{f}^{(i)}, \mathbf{r}^{(i)}\right)$ and $\mathcal{S}_2^{(i)} \triangleq \left(\boldsymbol{\alpha}^{(i)}\right)$, and at the ith iteration, respectively. We now proceed by proposing Algorithm 1 to solve the problem (18).

4 | NUMERICAL SIMULATIONS

This section expresses the performance metrics and impacts studied through numerical simulations.

4.1 | Simulations setup

To understand the performance and resolution of the problem (17) for minimising the e2e latency, we relied on numerical simulations. The values of parameters used for attaining these results are summarised in Table 1.

4.2 | Results and discussion

4.2.1 | Convergence of the proposed algorithm

Figure 3 clearly demonstrates the convergence of the proposed algorithm in reducing the worse-case e2e latency. In particular, with the model of $M_k = 4$ UEs, $K = 2$ UAVs, the optimising process converges at six iterations. The figure additionally illustrates the impact of the required computation resource on the e2e latency. Unsurprisingly, when the required computation resource increases, the e2e latency of computational tasks gradually increases. For instance, the observed scenario witnesses a considerable rise in the worst-case e2e latency from 0.6 s to approximate 0.87 s when Cm rises to 1100 megacycles.

TABLE 1 Simulation parameters [48, 53, 55, 56]

Parameters	Value
Number of antennas	$L = 8$
Maximum transmit power	$P_m^{\max} = 23$ dBm
Bandwidth	$B = 5$ MHz
Transmission duration URLLC	$\delta = 0.02$ ms
Decoding error probability	$\varepsilon_{mk} = 10^{-5}$
Noise spectral density	-174 dBm/Hz
Maximum blocklength	$N = \tau B = 200$
Number of UEs	$M = \{4, 5, 6\}$
Number of ESs	$K = 2$
Maximum UEs' processing rate	$F_{\max}^{\text{loc}} = 1.5$ GHz
Maximum processing rate	$F_{\max}^{\text{es}} = 3$ GHz
Input data size	$D_{mk} = 100$ KB
Required computation resource	$C_{mk} = 1200 \times 10^6$ cycles
Total delay requirement	$T_m^{\max} = 2$ s
Minimum data rate	$R_{\min}^{ul} = 1$ Mbps
Maximum energy consumption UE	$E_m^{\max} = 0.5$ Joule
Maximum power consumption UE	$P_k^{\max} = 5$ W
Effective capacitance coefficient	$\theta_m = 10^{-27}$ W.s^3/cycle^3

Abbreviations: UE, user equipment; URLLC, ultra-reliable low-latency communications.

4.2.2 | Impact of required computation resource

Figure 4 plots the impacts of required computing resources of the tasks (Cm) in the e2e latency of UEs under the proposed algorithm and other benchmark schemes. Particularly, when the tasks are more complicated, which require more computation resources, the e2e latency gradually increases. For instance, in the model of $M_k = 4$, $K = 2$ as observed in Figure 4, when C_m rises from 800 to 1, 200 megacycles, the worst-case latency obtained by using Algorithm 1 increases by approximate 300 ms. In addition, Figure 4 clearly demonstrates that our proposed algorithm is far better than all benchmark schemes. These results prove that joint optimisation of both communication and computation resources significantly improves the performance of MEC-based systems in reducing the e2e latency of UEs.

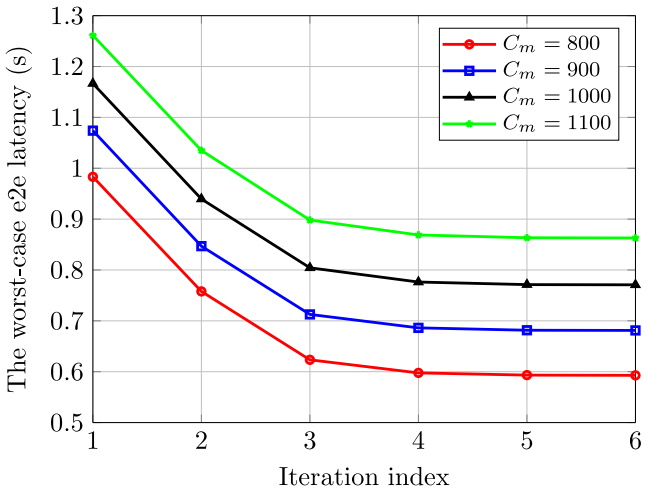

FIGURE 3 Convergence of the proposed algorithm with different values of required computation resource (C_m) in the scenarios of $M_k = 4$, $K = 2$

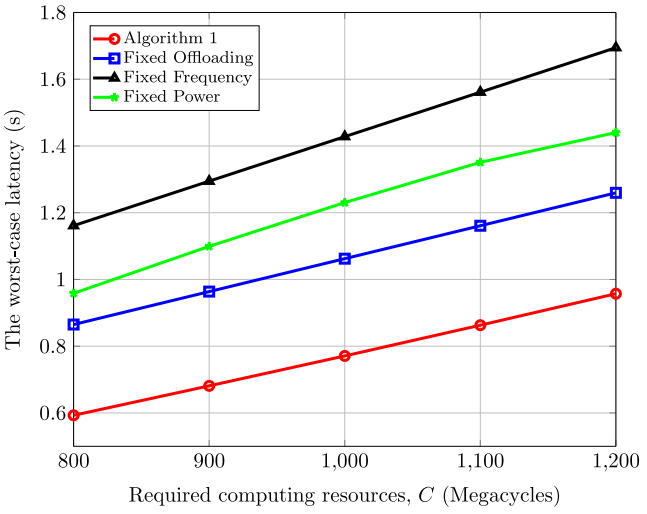

FIGURE 4 The worst-case latency among different values of required computation resource ($C_m \triangleq C$) in the scenarios of $M_k = 4$, $K = 2$

4.2.3 | Impact of UE transmit power budget

To investigate the impact of UEs transmit power in reducing the e2e latency, we have conducted experiments among different values of UE power budget in three models of $M_k = \{4, 5, 6\}$ and $K = 2$ UAVs. Figure 5 clearly states that when the transmit power budget of UEs increases, the worst-case e2e latency of UEs gradually reduces. For instance, the model of $M_k = 4$ UEs, $K = 2$ UAVs witnesses a considerable decline in latency by around 100 ms when the maximum of UEs transmit power reaches 23 dBm.

4.2.4 | Impact of UEs' processing rate

To demonstrate the impact of UEs' processing rate on obtaining the minimised latency and adjusting optimal offloading decisions, Figure 6 presents the numerical results of experiments with a range values of $F^{\text{loc}}_{\max}$. Unsurprisingly, when the processing capacity of UEs increases from 1 to 1.3 GHz, the e2e latency of UEs significantly reduces in both models. This is because the UEs are more powerful in processing tasks locally. Additionally, due to the constraints on UEs energy consumption in (17d), the average offloading portions of UEs have to increase to satisfy the energy budget when UEs have higher processing rate.

4.2.5 | Impact of ESs' processing rate

Figure 7 illustrates the impact of the total computation resource of ESs, $F^{\text{es}}_{\max}$ on the e2e latency of computational tasks coming from UEs. The figure clearly shows that the more powerful the ESs are, the less e2e latency can be achieved for both examined scenarios of $M_k = \{4, 6\}$ UEs, $K = 2$ UAVs. Figure 7 also indicates under the same computation resource budget of ESs, the model that has more UEs ($M_k = 6$) obtains higher e2e latency than that in the smaller size model ($M_k = 4$).

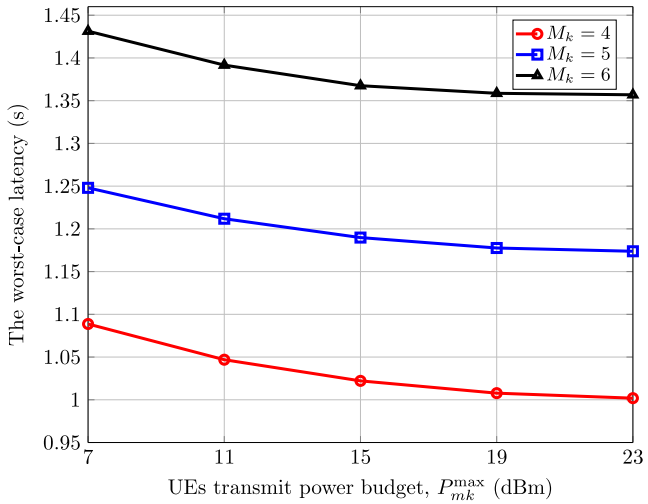

FIGURE 5 The worst-case latency among different values of user equipment (UE) transmit power budget $\left(P^{\max}_{mk}\right)$ in the scenarios of $M_k = 4$, $K = 2$, $M_k = 5$, $K = 2$, $M_k = 6$, $K = 2$

These results definitely demonstrate the effectiveness of the proposed offloading design.

5 | CONCLUSION

In conclusion, this paper has tackled the problem of minimising e2e latency of DT-assisted UAV-based edge networks. The latency minimisation problem has been successfully resolved through the AO-IA framework that is comprised of two sub-problems, namely optimal resource allocation and optimal tasks offloading policies. The numerical results have clearly demonstrated the effectiveness of the proposed solution. In the future, the model can be extended to include a multi-UAVs scenario with heterogeneous requests and offloading operations.

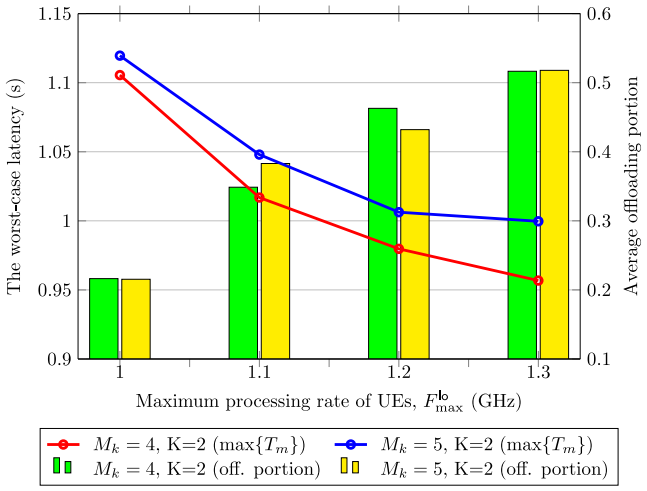

FIGURE 6 The worst-case latency among the maximum values of UEs' processing rate $\left(F^{\text{loc}}_{\max}\right)$ in the scenarios of $M_k = 4$, $K = 2$ and $M_k = 5$, $K = 2$

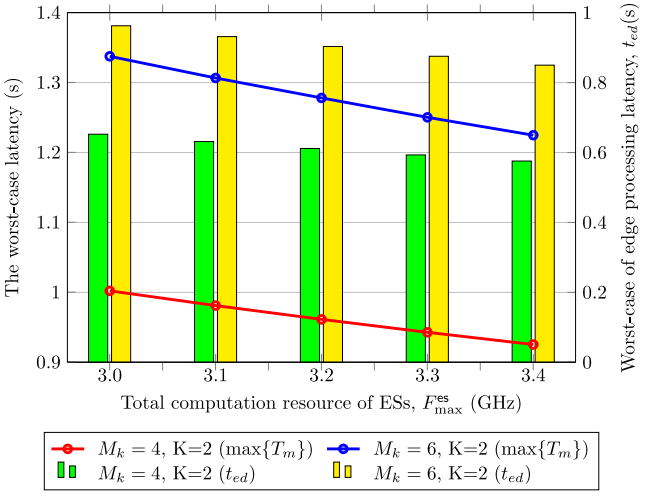

FIGURE 7 The worst-case latency among different values of total ESs' processing rate $\left(F^{\text{es}}_{\max}\right)$ in the scenarios of $M_k = 4$, $K = 2$ and $M_k = 6$, $K = 2$, $P^{\max}_k = 8$ W

ACKNOWLEDGEMENTS

This work has been supported in part by EPSRC-IAA (Grant / Award Number: '2020-2021'), Engineering and Physical Sciences Research Council (Grant/Award Number: 'EP/P019374/1'), and the U.K. Royal Academy of Engineering (RAEng) under the RAEng Research Chair and Senior Research Fellowship scheme (Grant RCSRF2021\11\41).

CONFLICT OF INTEREST

There is no conflict of interest.

DATA AVAILABILITY STATEMENT

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

ORCID

Dang Van Huynh https://orcid.org/0000-0002-2314-4934
Trung Q. Duong https://orcid.org/0000-0002-4703-4836

REFERENCES

1. Duong, T.Q., et al.: Digital twin-enabled 6G aerial edge computing with ultra-reliable and low-latency communications. In: Proc. 1st International Conference on 6G Networking. Paris, France (2022). (invited paper)
2. Wu, Y., Zhang, K., Zhang, Y.: Digital twin networks: a survey. IEEE Internet Things J. 8(18), 13789–13804 (2021)
3. Liu, T., et al.: Digital twin assisted task offloading based on edge collaboration in the digital twin edge network. IEEE Internet Things J. 4662(c), 1 (2021)
4. Sun, W., et al.: Reducing offloading latency for digital twin edge networks in 6G. IEEE Trans. Veh. Technol. 69(10) (2020)
5. Lopez, J., Rubio, J.E., Alcaraz, C.: Digital twins for intelligent authorization in the B5G-enabled smart grid. IEEE Wireless Commun. Mag. 28(2), 48–55 (2021)
6. Huynh, D.V., et al.: Digital twin empowered ultra-reliable and low-latency communications-based edge networks in industrial IoT environment. In: Proc. IEEE Int. Conf. Commun. (ICC'22). Seoul, Korea (2022)
7. Do-Duy, T., et al.: Digital twin-aided intelligent offloading with edge selection in mobile edge computing. IEEE Wireless Commun. Lett. (2022). accepted
8. Nguyen, M.-N., et al.: Real-time optimal resource allocation for embedded UAV communication systems. IEEE Wireless Commun. Lett. 8(1), 225–228 (2018)
9. Nguyen, L.D., Kortun, A., Duong, T.Q.: An introduction of real-time embedded optimisation programming for UAV systems under disaster communication. EAI Endorsed Trans. Ind. Netw. Intell. Syst. 5(17), e5 (2018)
10. Shin Kang, Y., et al.: Ground test results of flight control system for the Smart UAV. In: Proc. Int. Conf. Control, Autom. Syst., pp. 2533–2536. Gyeonggi-do, Korea (2010)
11. shin Kang, Y., et al.: Flight test of flight control performance for airplane mode of Smart UAV. In: Proc. 12th Int. Conf. Control, Autom. Syst., pp. 1738–1741. Guangzhou, China (2012)
12. Yoo, C.-S., et al.: Collision avoidance of Smart UAV in multiple intruders. In: Proc. 12th Int. Conf. Control, Autom. Syst., pp. 443–447. Guangzhou, China (2012)
13. Hua, M., et al.: UAV-assisted Intelligent Reflecting Surface symbiotic radio system. IEEE Trans. Wireless Commun. (2021)
14. Xu, J., et al.: Autonomous decision-making method for combat mission of UAV based on Deep Reinforcement Learning. In: Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC), vol. 1, pp. 538–544. Chengdu, China (2019)
15. Wang, Y., Wang, H., We, X.: Energy-efficient UAV deployment and task scheduling in multi-UAV edge computing. In: Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP), pp. 1147–1152. Wuhan Hubei, China (2020)
16. Nguyen, L.D., et al.: Real-time deployment and resource allocation for distributed UAV systems in disaster relief. In: Proc. IEEE 20th Int. Workshop Signal Process. Adv. Commun. (SPAWC), pp. 1–5. Cannes, France (2019)
17. Do, H.T., et al.: Formation control algorithms for multiple-UAVs: a comprehensive survey. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems. 8(27), 6 (2021)
18. Li, S., et al.: Unsupervised detection of earthquake-triggered roof-holes from UAV images using joint color and shape features. IEEE Geosci. Rem. Sens. Lett. 12(9), 1823–1827 (2015)
19. Mohammed, F., et al.: UAVs for smart cities: opportunities and challenges. In: Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS), pp. 267–273. Orlando, FL USA (2014)
20. Shao, X., et al.: A unified design of massive access for cellular internet of things. IEEE Internet Things J. 6(2), 3934–3947 (2019)
21. Yang, X., et al.: High accuracy active stand-off target geolocation using UAV platform. In: Proc. IEEE Int. Conf. Signal, Inf. Data Process. (ICSIDP), pp. 1–4. Chongqing, China (2019)
22. Waharte, S., Trigoni, N.: Supporting search and rescue operations with UAVs. In: Proc. Int. Conf. Emerg. Secur. Technol. pp. 142–147. Canterbury, United Kingdom (2010)
23. Wang, S., et al.: A Deep-Learning-Based sea search and rescue algorithm by UAV remote sensing. In: Proc. IEEE CSAA Guid. Navigat. Control Conf. (CGNCC), pp. 1–5. Xiamen, China (2018)
24. Do-Duy, T., et al.: Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications. IEEE J. Sel. Area. Commun. (2021)
25. Sutton, G.J., et al.: Enabling technologies for ultra-reliable and low latency communications: from PHY and MAC layer perspectives. IEEE Commun. Surveys Tuts. 21(3), 2488–2524 (2019)
26. Yoshizawa, T., Baskaran, S.B.M., Kunz, A.: Overview of 5G URLLC system and security aspects in 3GPP. In: Proc. IEEE Conf. Standards Commun. Netw. (CSCN), pp. 1–5. Granada, Spain (2019)
27. Durisi, G., Koch, T., Popovski, P.: Toward massive, ultra-reliable, and low-latency wireless communication with short packets. Proc. IEEE. 104(9), 1711–1726 (2016)
28. Wu, Y., et al.: Secrecy-based delay-aware computation offloading via mobile edge computing for internet of things. IEEE Internet Things J. 6(3), 4201–4213 (2018)
29. Mu, N., et al.: The 5G MEC applications in smart manufacturing. In: Proc. IEEE Int. Conf. Edge Comput. (EDGE), Virtual Conference, pp. 45–48 (2020)
30. Rafiq, A., et al.: Optimizing energy consumption and latency based on computation offloading and cell association in MEC enabled Industrial IoT environment. In: Proc. 6th Int. Conf. Intell. Comput. Signal Process. (ICSP), pp. 10–14. Xi'an, China (2021)
31. Ha, D.-B., Truong, V.-T., Lee, Y.: Performance analysis for rf energy harvesting mobile edge computing networks with SIMO/MISO-NOMA schemes. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems. 8(27) (2021)
32. Kherani, A.A., et al.: Development of MEC system for indigenous 5G test-bed. In: Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS), pp. 131–133. Bangalore, India (2021)
33. Li, J., et al.: Deep Reinforcement Learning based computation offloading and resource allocation for MEC. In: Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), pp. 1–6. Barcelona, Spain (2018)
34. Wu, J., et al.: Edge-cloud collaborative computation offloading model based on improved partical swarm optimization in MEC. In: Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS), pp. 959–962. Tianjin, China (2019)
35. Altin, İ., Akar, M.: Novel OMA and hybrid NOMA schemes for MEC offloading. In: Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom), pp. 1–5. Odessa, Ukraine (2020)

36. Gonzalez-Sosa, E., et al.: Audience meter: a use case of deploying machine learning algorithms over 5G networks with MEC. In: Proc. IEEE Int. Conf. Consumer Electron. (ICCE), pp. 1–2. Phu Quoc Island, Vietnam (2020)
37. Yang, Y., Hu, Y., Gursoy, M.C.: Reliability-optimal designs in MEC networks with finite blocklength codes and outdated CSI. In: Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS), Virtual Conference, pp. 1–6 (2021)
38. Tan, G., Zhang, H., Zhou, S.: Resource allocation in MEC-enabled vehicular networks: a Deep Reinforcement Learning approach. In: Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), Virtual Conference, pp. 406–411 (2020)
39. Lee, S., Lee, S., Shin, M.-K.: Low cost MEC server placement and association in 5G networks. In: Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), pp. 879–882. Jeju-si and Jeju-do, South Korea (2019)
40. Jia, Z., et al.: 5G MEC gateway system design and application in industrial communication. In: Proc. 2nd World Symp. Artif. Intell. (WSAI), pp. 5–10. Guangzhou, China (2020)
41. Hao, M., et al.: URLLC resource slicing and scheduling in 5G vehicular edge computing. In: Proc. IEEE 93rd Veh. Technol. Conf. (VTC2021-Spring), pp. 1–5. Guangzhou, China (2021)
42. Zhan, C., et al.: Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications. IEEE Internet Things J. 8(20), 15553–-15567 (2021)
43. Zhou, F., et al.: Mobile edge computing in unmanned aerial vehicle networks. IEEE Wireless Commun. 27(1), 140–146 (2020)
44. Xu, D., et al.: Robust resource allocation for UAV systems with UAV jittering and user location uncertainty. In: Proc. 2018 IEEE Globecom Workshops (GC Wkshps). Abu Dhabi, United Arab Emirates (2018)
45. Xu, D., et al.: Multiuser MISO UAV communications in uncertain environments with no-fly zones: robust trajectory and resource allocation design. IEEE Trans. Commun. 68(5), 3153–3172 (2020)
46. Tun, Y.K., et al.: Energy-efficient resource management in UAV-assisted mobile edge computing. IEEE Commun. Lett. 25(1), 249–253 (2021)
47. Zhou, F., et al.: Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. IEEE J. Sel. Area. Commun. 36(9), 1927–1941 (2018)
48. Nasir, A.A.: Latency optimization of UAV-enabled MEC system for virtual reality applications under Rician fading channels. IEEE Wireless Commun. Lett. 10(8), 1633–1637 (2021)
49. Bennis, M., Debbah, M., Poor, H.V.: Ultrareliable and low-latency wireless communication: tail, risk, and scale. Proc. IEEE. 106(10), 1834–1853 (2018)
50. Elbamby, M.S., et al.: Wireless edge computing with latency and reliability guarantees. Proc. IEEE. 107(8), 1717–1737 (2019)
51. Pan, Y., et al.: UAV-assisted and intelligent reflecting surfaces-supported terahertz communications. IEEE Wireless Commun. Lett. 10(6), 1256–1260 (2021)
52. Ren, H., et al.: Joint pilot and payload power allocation for massive-MIMO-enabled URLLC IIoT networks. IEEE J. Sel. Area. Commun. 38(5), 816–830 (2020)
53. She, C., Yang, C., Quek, T.Q.S.: Radio resource management for ultra-reliable and low-latency communications. IEEE Commun. Mag. 55(6), 72–78 (2017)
54. Liu, Q., Han, T., Ansari, N.: Joint radio and computation resource management for low latency mobile edge computing. In: Proc. IEEE Global Commun. Conf., GLOBECOM 2018. Abu Dhabi, United Arab Emirates (2018)
55. Wang, J., et al.: Joint computation offloading and resource allocation for MEC-enabled IoT systems with imperfect CSI. IEEE Internet Things J. 8(5), 3462–3475 (2021)
56. Liu, C.-F., et al.: Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans. Commun. 67(6), 4132–4150 (2019)
57. Do-Duy, T., et al.: Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications. IEEE J. Sel. Area. Commun. 39(11), 3411–3424 (2021)
58. Nguyen, L.D., et al.: Real-time deployment and resource allocation for distributed UAV systems in disaster relief. In: Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), pp. 1–5. Cannes, France (2019)
59. Duong, T.Q., Nguyen, L.D., Nguyen, L.K.: Practical optimisation of path planning and completion time of data collection for UAV-enabled disaster communications. In: Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), pp. 372–377. Tangier, Morocco (2019)
60. Marks, B.R., Wright, G.P.: A general inner approximation algorithm for nonconvex mathematical programs. Oper. Res. 26(4), 681–683 (1978)
61. Beck, A., Ben-Tal, A., Tetruashvili, L.: A sequential parametric convex approximation method with applications to nonconvex truss topology design problems. J. Global Optim. 47(1), 29–51 (2010)
62. Nasir, A.A., et al.: Resource allocation and beamforming design in the short blocklength regime for URLLC. IEEE Trans. Wireless Commun. 20(2), 1321–1335 (2021)
63. Grant, M., Boyd, S.: CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1 (2014). http://cvxr.com/cvx

How to cite this article: Li, Y., et al.: Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: a digital twin approach. IET Signal Process. 16(8), 897–908 (2022). https://doi.org/10.1049/sil2.12128

APPENDIX

We first rewrite the SINR of UE (m, k) as $\gamma_{mk}(\mathbf{p}) = p_{mk}/q_{mk}(\mathbf{p})$. By applying the inequality [63, Equation (72)] for $x = p_{mk}$, $y = q_{mk}(\mathbf{p})$, $\overline{x} = p_{mk}^{(i)}$, and $\overline{y} = q_{mk}(\mathbf{p}^{(i)})$, we have

$$G_{mk}(\mathbf{p}) \geq a_{mk}^{(i)} - \frac{b_{mk}^{(i)}}{p_{mk}} - c_{mk}^{(i)} q_{mk}(\mathbf{p}) \triangleq \mathcal{G}_{mk}^{(i)}(\mathbf{p})$$

where $a_{mk}^{(i)} = \ln\left(1 + \frac{p_{mk}^{(i)}}{q_{mk}^{(i)}(\mathbf{p})}\right) + 2\frac{p_{mk}^{(i)}}{p_{mk}^{(i)}+q_m^{(i)}(\mathbf{p})}$, $b_{mk}^{(i)} = \frac{\left(p_{mk}^{(i)}\right)^2}{p_{mk}^{(i)}+q_{mk}^{(i)}(\mathbf{p})}$ and $c_{mk}^{(i)} = \frac{p_{mk}^{(i)}}{\left(q_{mk}^{(i)}(\mathbf{p})+p_{mk}^{(i)}\right)q_{mk}^{(i)}(\mathbf{p})}$.

To find an upper bounding convex function approximation of $W_{mk}(\mathbf{p})$, we apply the inequality [63, Eq. (75)] for $x = 1 - 1/(1+\gamma_{mk}(\mathbf{p}))^2$ and $\overline{x} = 1 - 1/\left(1+\gamma_{mk}\left(\mathbf{p}^{(i)}\right)\right)^2$, yielding

$$W_{mk}\left(\mathbf{p}, \boldsymbol{\pi}_{mk}^{(i)}\right) \leq = d_{mk}^{(i)} - e_{mk}^{(i)} \frac{q_{mk}^2(\mathbf{p})}{\left(q_{mk}(\mathbf{p}) + p_{mk}\right)^2} \tag{A1}$$

where

$$d_{mk}^{(i)} = 0.5\sqrt{V_{mk}(\mathbf{p}^{(i)})} + 0.5/\sqrt{V_{mk}(\mathbf{p}^{(i)})} \tag{A2}$$

$$e_{mk}^{(i)} = 0.5/\sqrt{V_{mk}(\mathbf{p}^{(i)})}. \tag{A3}$$

The function $\frac{q_{mk}^2(\mathbf{p})}{\left(q_{mk}(\mathbf{p})+p_{mk}\right)^2}$ in (A1) is still not convex [63], which can be further approximated by using inequalities [63, Eq. (77)] and [63, Eq. (76)] as

$$\frac{q_{mk}^2(\mathbf{p})}{q_{mk}(\mathbf{p})+p_{mk}}\frac{1}{q_{mk}(\mathbf{p})+p_{mk}}\geq\frac{2}{q_{mk}(\mathbf{p}^{(i)})+p_{mk}^{(i)}}\times\left(\frac{2q_{mk}(\mathbf{p}^{(i)})q_{mk}(\mathbf{p})}{q_{mk}(\mathbf{p}^{(i)})+p_{mk}^{(i)}}-\frac{q_m^2(p_m^{(i)})}{\left(q_m\left(p_m^{(i)}\right)+p_m^{(i)}\right)^2}(q_{mk}(\mathbf{p})+p_{mk})\right)-\frac{q_{mk}^2(\mathbf{p})}{\left(q_{mk}^{(i)}(\mathbf{p})+p_{mk}^{(i)}\right)^2} \tag{A4}$$

over the trusted regions defined in (22) and (23). By substituting this result to (A1) yields

$$\begin{aligned}W_{mk}\left(\mathbf{p},\boldsymbol{\pi}_{mk}^{(i)}\right)&\leq\mathcal{W}_m^{(i)}(\mathbf{p})\\&\triangleq d_{mk}^{(i)}-\frac{2e_{mk}^{(i)}}{q_{mk}(\mathbf{p}^{(i)})+p_{mk}^{(i)}}\times\left(2f_{mk}^{(i)}q_{mk}(\mathbf{p})-\left(f_{mk}^{(i)}\right)^2(q_{mk}(\mathbf{p})+p_{mk})\right)\\&\quad+\frac{\left(f_{mk}^{(i)}\right)^2}{q_{mk}^2(\mathbf{p}^{(i)})}q_{mk}^2(\mathbf{p}),\end{aligned} \tag{A5}$$

where $f_{mk}^{(i)}\triangleq\frac{q_{mk}(\mathbf{p}^{(i)})}{q_{mk}(\mathbf{p}^{(i)})+p_{mk}^{(i)}}$.