Distributed Communication and Computation Resource Management for Digital Twin-aided Edge Computing with Short-Packet Communications

Dang Van Huynh, *Graduate Student Member, IEEE,* Van-Dinh Nguyen, *Member, IEEE,* Saeed R. Khosravirad, *Member, IEEE,* George K. Karagiannidis, *Fellow, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—For future networks, it is highly demanding to satisfy a wide range of time-sensitive and computation-intensive services. This is a very challenging task, since it requires a combination of aspects from information, communication and computation in order to establish a digital representation of the real network environment. This paper introduces a fairness-aware latency minimisation (FALM) framework in the digital twin (DT) aided edge computing with ultra-reliable and low latency communications (URLLC), which jointly optimises various communication and computation parameters, namely, bandwidth allocation, transmission power, task offloading portions, and processing rate of user equipments (UEs) and edge servers (ESs). The formulated problem is highly complicated, due to non-convex constraints and strong coupling among optimisation variables. To deal with this problem, we develop both centralised and distributed optimisation approaches. In particular, we first resort to successive convex approximation (SCA) method to develop a low-complexity iterative algorithm and solve the problem in a centralised manner. Combining tools from SCA and alternating direction method of multipliers (ADMM), we develop an efficient distributed solution with parallel computation processing at ESs under global consensus in each iteration and strong theoretical performance guaranteed. Numerical results are provided to validate the proposed solutions in terms of convergence speed and overall latency as well as improving fairness among all UEs.**

***Index Terms*—Digital twin, distributed optimisation, industrial Internet of Things, edge computing, ultra-reliable and low latency communications.**

D. V. Huynh and T. Q. Duong are with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, BT7 1NN, UK. (e-mail: {dhuynh01, trung.q.duong}@qub.ac.uk).

V.-D. Nguyen is with the College of Engineering and Computer Science and also with the Center for Environmental Intelligence, VinUniversity, Vinhomes Ocean Park, Hanoi 100000, Vietnam (e-mail: dinh.nv2@vinuni.edu.vn).

S. R. Khosravirad is with Nokia Bell Labs, Murray Hill, NJ 07964 USA (e-mail: saeed.khosravirad@nokia-bell-labs.com).

G. K. Karagiannidis is with the Wireless Communications and Information Processing (WCIP) Group, Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, Thessaloniki 54 124, Greece (e-mail: geokarag@auth.gr).

This work was supported in part by the U.K. Royal Academy of Engineering (RAEng) under the RAEng Research Chair and Senior Research Fellowship scheme Grant RCSRF2021\11\41 and supported in part by the UK Department for Science, Innovation and Technology under the Future Open Networks Research Challenge project TUDOR (Towards Ubiquitous 3D Open Resilient Network). The work of V.-D. Nguyen was supported by the VinUniversity Seed Grant Program.

Corresponding author is Trung Q. Duong. This paper has been accepted in part for presentation at the 56th Annual Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, October 30 – November 2, 2022 [1].

I. Introduction

Ultra-reliable and low latency communications (URLLC) and edge computing have emerged as key technologies to enable a wide range of time-sensitive and computation-intensive applications such as smart factories, extended reality (XR) and vehicular edge computing (VEC) [2]–[4]. These technologies lie at the heart of the fifth-generation (5G) and beyond 5G wireless systems in terms of communication and computation perspectives. According to the 3GPP Releases 15 and 16, URLLC is being enhanced to achieve the stringent requirements of "five-nine" to "seven-nine", i.e. 99.99999% in reliability while ensuring end-to-end (e2e) latency on the order of 1 ms [5], [6]. Due to the highly complicated relationship between the transmission error probability and the delay in the finite blocklength regime, recent studies in URLLC mostly focused on resource allocation [7], [8], beamforming design [9] and reliability maximisation [10], [11]. Meanwhile, there are only a few attempts to jointly optimise URLLC and edge computing and meet the unprecedented demand of time-sensitive services [12], [13].

From a computational perspective, multi-access edge computing (MEC) provides a powerful framework to leverage the computing capacities of nearby edge servers (ES) or fog-cloud servers to reduce to processing time of computational tasks [14]. MEC is capable of supporting a wide range of computation-intensive applications that demand high quality-of-service (QoS) and the quality-of-experience (QoE) in the next generation of wireless systems [15]. The studies in MEC span in various practical problems, including task offloading, task caching, resource management, and federated learning [16], [17], that aims to minimise processing latency [18] and system cost [19], and to maximise energy efficiency [20]. A joint optimisation of communication and computation resources in MEC combined with critical wireless technologies (e.g. URLLC) has attached much attention from the research community but also opens many research challenges [3], [15].

This paper investigates the resource management problem of the distributed edge computing system subject to stringent requirements of URLLC-based transmissions. The digital twin (DT) concept is exploited to model the computation capacity of UEs and ESs. Both centralised and distributed approaches are employed to deal with the fairness-ware latency minimisation problem.

A. Literature Review

Task offloading has emerged as a critical technique for MEC that aims to transfer computationally intensive tasks to external servers equipped with more powerful computing capabilities. The constrained IoT devices can not only partially execute computation-intensive tasks but also offload a part of tasks to nearby ESs to optimise the processing time and maximise energy efficiency [20]–[22]. In particular, a relaxation-based convex optimisation algorithm was developed to minimise the energy consumption by jointly optimising offloading decision and resource allocation in [20]. A low-complexity framework to effectively solve the mixed-integer non-convex problem of energy consumption minimisation was investigated in [21]. Typically, the real-world deployments of MEC systems consist of multiple ESs to assist task processing. Therefore, the studies exploiting distributed optimisation approach in MEC architecture are recently attracting much attention [23], [24]. Specifically, a distributed algorithm based on branch-and-bound approach was provided to optimise the workload offloading values of IoT devices and minimise the delay of task processing in [23]. Two distributed optimisation algorithms based on the subgradient method and alternating direction method of multipliers (ADMM) to minimise the overall delay were proposed in [24].

Moreover, DT has emerged as a promising industrial paradigm that can fully replicate the physical devices and produce real-time interactions to efficiently manage the entire system. Recently, it has been shown that DT has potential to provide solutions to duplicate the physical networked systems so that the network resources can be effectively managed [25]. The DT edge network has been considered in [26] that aims to minimise the offloading latency by an actor-critic deep reinforcement learning (DRL) algorithm. In [27], the DRL-based approach was also developed to solve the problem of task offloading and edge selection in DT-assisted edge networks. More recently, the DT concept was applied in modelling the dynamic computation allocation for UEs and ES to reduce the e2e latency of IoT applications [28].

On the other hand, URLLC is one of the key pillars of 5G New Radio (NR) that helps to achieve shorter transmissions through a larger subcarrier, addressing diverse mission-critical applications such as industrial automation, intelligent transportation and tactile Internet, etc [29]. The researches on URLLC have mainly focused on the resource allocation [7], [8], [30] and beamforming design [9] for short-packet communications. In particular, the works in [8] and [30] investigated the radio resource management for URLLC to optimise the resource usage and energy efficiency. Joint pilot and payload power allocation for massive multiple-input multiple-output (MIMO) was considered in [7] to maximise the weighted sum rate of all devices in industrial scenarios. The authors in [9] developed low-complexity path-following algorithms to solve a joint optimisation of the resource allocation and beamforming design in the short-packet regime for a downlink URLLC system, where newly approximate convex functions are derived to convexify the complicated rate function of of bandwidth, transmission power and beamforming vectors.

It is expected that URLLC can bring great benefits to MEC to enable new types of time-sensitive services, such as virtual reality/augmented reality (VR/AR) and metaverse applications [12], [31]–[33]. Towards an energy-efficient solution, the dynamic computation offloading problem in MEC via URLLC links was considered in [31] by jointly optimising energy consumption of users and computational resources of ESs. In addition, the application of DT concept was exploited in [12] to find the optimal resource allocation and offloading probabilities based on the learning-aided approach. More recently, the joint design of MEC, URLLC and DT has been investigated in [32]–[34] by taking into account the impacts of user association, offloading portions, transmission power and processing rate of UEs and ESs. Nevertheless, these works mainly focused on developing a centralised solution, and thus the computation ability of ESs is not fully utilised.

B. Motivation and Contributions

The use of URLLC and MEC to realise the full potential of DT is still in the early stage. This requires a comprehensive optimisation design of both communication and computation resources as well as exploiting the powerful ability of distributed edge computing. On the one hand, the DT's application in MEC investigated in [12], [27] and [28] has mainly focused on the task offloading optimisation while other communication factors are not fully taken into account. It is noted that in the DT networks, communication and computation resources are highly dependent which may cause correlated failures and overloads of edge nodes. On the other hand, ESs are often deployed in widely distributed geographies, which promotes the distributed solution that can reduce not only information exchange between nodes but also the *round-trip* time. In addition, a fairness design has recently become more critical in wireless networks to improve network utilisation and UEs' QoE. However, to the best of our knowledge, the previous works (e.g. [21], [31]–[34]) neither consider a distributed solution to enable parallel computation processing at ESs nor simultaneously guarantee fairness among all UEs. This calls for an efficient distributed solution to fully exploit the computing power of edge nodes while still guaranteeing comparable fairness among all individual UEs.

In this paper, we propose a new optimisation framework to minimise the overall end-to-end (e2e) latency of UEs in the DT-aided edge computing with URLLC, taking into account all the issues mentioned above. Both centralised and distributed optimisation approaches are developed to solve the joint optimisation problem of communication and computation resources.

The main contributions of this paper can be summarised as follows:

- We first formulate a fairness-aware latency minimisation (FALM) problem for DT networks by jointly optimising offloading portions, processing rates of UEs and ESs, bandwidth allocation and transmit power of UEs. Notably, we introduce a fairness parameter ($q \geq 0$) into the objective function to effectively and flexibly improve fairness among all UEs, without the need of complicated function or additional constraints.

- We propose a low-complexity iterative algorithm to solve the FALM problem in a centralised manner. In particular, we first apply the successive convex approximation (SCA) method to convexify the nonconvex constraints and the develop an iterative algorithm to successively solve the approximate convex program.
- Given the insights from the centralised approach, the next step is to construct a distributed solution to fully exploit the parallel computation processing at ESs. Since the approximate convex program obtained by SCA is in a standard form for a direct application of the ADMM method, we introduce new local and global variables to transform the original optimisation problem into the separable convex subproblems which can be solved independently at each ES. After each ADMM iteration, all ESs update the involved parameters to form the next approximate convex program. The distributed solution achieves the same performance as the centralised approach.
- Finally, we numerically evaluate the performance of the proposed algorithms in terms of the convergence speed, fairness analysis and consensus evolution under various impacts of the resource budgets. The results confirm significant performance improvement of the proposed algorithms, compared to the existing schemes. They also reveal that the proposed scheme can reduce latency by $33-66\%$ compared from baselines and achieve fairness up to 99.6% based on Jain's index.

C. Paper Structure and Notations

The rest of this paper is structured as follows. Section II describes the system model and formulates the FALM problem. We provide the centralised solution in Section III, while the distributed optimisation approach is given in Section IV. The complexity analysis and the convergence of the proposed algorithms are analyzed in Section V. Section VI provides the numerical results and discussions, and then Section VII concludes the paper.

Notation: Throughout the paper, vectors are denoted by bold lowercase letters. $\mathcal{CN}\left(\mu,\sigma^2\right)$ is circularly symmetric complex Gaussian random variable with mean μ and variance σ^2. $\mathbb{C}$ represents the space of complex matrices and vectors. $(\cdot)^*$ and $(\cdot)^H$ denote the conjugate of a complex number and the conjugate transpose of a matrix or vector, respectively.

II. System Model and Problem Formulation

A. DT-aided Distributed Edge Computing Model

Fig. 1 illustrates an DT-aided distributed edge computing architecture with URLLC for industrial automation. The physical system includes edge layer (e.g. industrial IoT UEs) and user layer (e.g. ESs). The connection between UEs and ESs is established by URLLC links to ensure stringent requirements on reliability and latency communications in industrial automation scenarios. Each UE can offload computational tasks to multiple ESs. The DT system provides services that aims to replicate the physical objects to perform estimations, optimisation, and making decisions to manage and control the physical system more effectively.

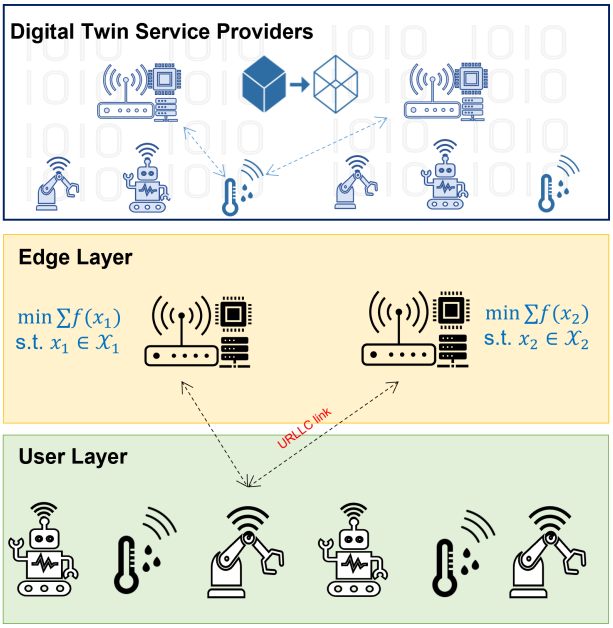


Fig. 1: Illustration of the DT-aided distributed edge computing with URLLC.

1) Task Offloading Model: We consider that there are the sets $\mathcal{M}=\{1,2,\cdots,M\}$ and $\mathcal{K}=\{1,2,\cdots,K\}$ of $M=|\mathcal{M}|$ UEs and $K=|\mathcal{K}|$ ESs, respectively. Each ES is associated with an access point (AP) for wireless communications. A task coming from UE $m\in\mathcal{M}$ is characterised by a tuple of $J_m=\{T_m^{\max},C_m,D_m\}$, where $T_m^{\max}$ is the maximum latency requirement (s), C_m is required CPU cycles to execute the whole task (cycles), and D_m is the task size (bits). Let α_m and β_{mk} be the portions of the task J_m executed locally at UE m and the offloading portion to ES $k\in\mathcal{K}$, respectively. For the task J_m, we can show that $D_m=\alpha_m D_m+\sum\limits_{k\in\mathcal{K}}\beta_{mk}D_m$, $C_m=\alpha_m C_m+\sum\limits_{k\in\mathcal{K}}\beta_{mk}C_m$ and $\alpha_m+\sum\limits_{k\in\mathcal{K}}\beta_{mk}=1,\forall m$. Let us define $\boldsymbol{\alpha}\triangleq\{\alpha_m\}_{\forall m}$ and $\boldsymbol{\beta}\triangleq\{\beta_{mk}\}_{\forall m,k}$.

2) DT Model: In DT services, all the physical devices (e.g. UEs and ESs) are fully replicated in terms of hardware configuration, software settings and working states. DT services can powerfully make decisions to manage and control the entire physical system in real-time to guarantee the performance. To do this, the DT model of the distributed URLLC-based edge computing can be expressed as

$$\mathsf{DT}=\{\tilde{\mathcal{M}},\tilde{\mathcal{K}}\} \tag{1}$$

where $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{K}}$ denote the digital representations of UEs and ESs, respectively. Based on real-time interactions with the physical objects, the DT service promptly provide the optimal solutions on tasks offloading, estimated processing rate, bandwidth allocation, and transmission power to optimise the performance of the entire system.

The DT service for UE m can be modelled as

$$\mathsf{DT}_m=(f_m^{\mathsf{ue}},\hat{f}_m^{\mathsf{ue}}) \tag{2}$$

where f_m^{ue} is the estimated processing rate of the m-th UE, and $\hat{f}_m^{\mathsf{ue}}$ is the deviation between the physical device and its DT. Similarly, the DT model of ES k (denoted by DT_k) can

be expressed as

$$\mathsf{DT}_k = (f_{mk}^{\mathsf{es}}, \hat{f}_{mk}^{\mathsf{es}}) \tag{3}$$

where f_{mk}^{es} is the estimated processing rate of the k-th physical ES to handle the task from m-th UE, and $\hat{f}_{mk}^{\mathsf{es}}$ is the deviation between the physical ES and its DT. The DT services optimise the estimated processing rate of ESs to reflect current configuration of the physical ESs in terms of computing ability. This mechanism allows the DT to make decisions on adjusting offloading factors and the processing rate of ESs to maximise the system performance.

B. URLLC-based Transmission Model

Since the data size of the computation results is usually small and APs are equipped with more powerful computing power than UEs, the downlink transmission latency can be ignored [28], [35]. The system bandwidth is B, and each AP is equipped with $L > 1$ antennas while each UE has single antenna. We adopt the frequency division multiple access (FDMA) protocol, where the portion of bandwidth allocated to the m-th UE by the k-th AP is b_{mk}, satisfying:

$$\sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}} b_{mk} \leq 1. \tag{4}$$

The channel vector between UE m and AP k is denoted by $\mathbf{h}_{mk} \in \mathbb{C}^{L\times 1}$ that can be modelled as $\mathbf{h}_{mk} = \sqrt{g_{mk}}\bar{\mathbf{h}}_{mk}$, where g_{mk} is the large-scale channel coefficient including the path-loss and shadowing, and $\bar{\mathbf{h}}_{mk}$ is the small-scale fading following the Rayleigh fading model as $\bar{\mathbf{h}}_{mk} \sim \mathcal{CN}(0, \mathbf{I}_L)$. The $L \times 1$ received signal vector at the k-th AP can be expressed as

$$\mathbf{y}_k = \sum_{m\in\mathcal{M}} \mathbf{h}_{mk}\sqrt{p_{mk}}s_m + \mathbf{n}_k \tag{5}$$

where p_{mk} and s_m are the transmit power and unit-power data symbol of UE m, respectively; $\mathbf{n}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_L)$ is the additive white Gaussian noise (AWGN), where the variance of each element is normalized to unit. We note that the large-scale channel coefficient g_{mk} is normalized by the noise power.

Each coherence time is divided into two main phases, including n_p symbols for uplink training and n_d symbols for data transmission. The total number of symbols is given as $N = n_p + n_d$. We assume that all UEs send their pilot sequences to APs to perform channel estimation at the begining of each time slot. We assume that all M UEs share the same symbol duration for channel estimation and the minimum length of the pilot sequences is $n_p = M$ [7]. The minimum mean square error (MMSE) channel estimate of $\mathbf{h}_{mk}$ is given by [7]:

$$\hat{\mathbf{h}}_{mk} = \frac{g_{mk}Mp_{mk}^p}{g_{mk}Mp_{mk}^p + 1}\mathbf{y}_{mk}^p \tag{6}$$

where p_{mk}^p is the pilot transmit power of UE m. The estimated channel $\hat{\mathbf{h}}_{mk}$ follows the distribution of $\mathcal{CN}\left(\mathbf{0}, \sigma_{mk}^2\mathbf{I}\right)$, where σ_{mk}^2 is given as $\sigma_{mk}^2 = g_{mk}^2 Mp_m^p/(g_{mk}Mp_{mk}^p + 1)$. According to the MMSE estimation property, the channel estimation error $\tilde{\mathbf{h}}_{mk} = \mathbf{h}_{mk} - \hat{\mathbf{h}}_{mk}$ is independent of $\hat{\mathbf{h}}_{mk}$ that follows the distribution of $\mathcal{CN}\left(\mathbf{0}, \delta_{mk}^2\mathbf{I}_L\right)$, where δ_{mk}^2 is given by $\delta_{mk}^2 = g_{mk}/(g_{mk}Mp_{mk}^p + 1)$. Under FDMA, the bounded signal-to-noise ratio (SNR) of the m-th to the k-th AP can be calculated as

$$\gamma_{mk}(p_{mk}) = \frac{p_{mk}\left(L-1\right)\sigma_{mk}^2}{p_{mk}\delta_{mk}^2 + 1}. \tag{7}$$

Then, the approximation of the achievable transmission rate of the m-th UE to the k-th AP (bits/s) in the URLLC finite blocklength is given by [12], [36]:

$$R_{mk}(b_{mk}, p_{mk}) = \frac{(1-\omega_k)B}{\ln 2}\Big[b_{mk}\ln\big(1+\gamma_{mk}(p_{mk})\big) - \sqrt{\frac{b_{mk}V_{mk}(p_{mk})}{\phi B}}Q^{-1}(\epsilon_{mk})\Big] \tag{8}$$

where $\omega_k = M/N$, ϕ is the transmission time interval, ϵ_{mk} is the decoding error probability, $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp^{\left(\frac{-t^2}{2}\right)} dt$ is the inverse function, and $V_{mk}(p_{mk}) = 1 - \left[1+\gamma_{mk}(p_{mk})\right]^{-2}$ is the channel dispersion. As a results, the latency for task offloading from the m-th UE to the k-th AP can be expressed as

$$T_m^{\mathsf{co}}\left(b_{mk}, p_{mk}, \beta_{mk}\right) = \frac{\beta_{mk}D_m}{R_{mk}(b_{mk}, p_{mk})}. \tag{9}$$

C. DT-based Computation Model

1) Local Processing: UE m locally executes a portion α_m of the task with the estimated processing rate of f_m^{ue}. Then, the estimated time required to execute the task locally at UE is calculated as

$$\tilde{T}_m^{\mathsf{ue}}\left(\alpha_m, f_m^{\mathsf{ue}}\right) = \frac{\alpha_m C_m}{f_m^{\mathsf{ue}}}. \tag{10}$$

The deviation between the physical UE m and its DT can be acquired in advance [27], [28], the computing latency gap between the real value and DT estimation is given as

$$\Delta T_m^{\mathsf{ue}}\left(\alpha_m, f_m^{\mathsf{ue}}\right) = \frac{\alpha_m C_m \hat{f}_m^{\mathsf{ue}}}{f_m^{\mathsf{ue}}\left(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\right)} \tag{11}$$

where $\hat{f}_m^{\mathsf{ue}}$ is the deviation of the real processing rate. Consequently, the actual time for the local executing is given as

$$T_m^{\mathsf{ue}} = \Delta T_m^{\mathsf{ue}} + \tilde{T}_m^{\mathsf{ue}}. \tag{12}$$

We note that DTs typically try to estimate the processing rate as accurately as possible to control the system in real-time. Therefore, if the deviation in the processing rate estimation is large, the overall latency of the system will increase unexpectedly.

2) Edge Processing: ES k estimates the processing rate in the DT f_{mk}^{es} to handle the task offloaded from the m-th UE. The estimated latency of the k-th ES to execute the task J_m is given as

$$\tilde{T}_m^{\mathsf{es}}\left(\beta_{mk}, f_{mk}^{\mathsf{es}}\right) = \frac{\beta_{mk}C_m}{f_{mk}^{\mathsf{es}}}. \tag{13}$$

Next, the latency gap ΔT_m^{es} between the real value and DT estimation can be expressed as

$$\Delta T_m^{\mathsf{es}}\left(\beta_{mk}, f_{mk}^{\mathsf{es}}\right) = \frac{\beta_{mk}C_m\hat{f}_{mk}^{\mathsf{es}}}{f_{mk}^{\mathsf{es}}\left(f_{mk}^{\mathsf{es}} - \hat{f}_{mk}^{\mathsf{es}}\right)}. \tag{14}$$

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

$$T_m^{\mathsf{es}} = \Delta T_m^{\mathsf{es}} + \tilde{T}_m^{\mathsf{es}}. \tag{15}$$

D. Optimisation Problem Formulation

1) System Latency Model: The e2e latency of the task coming from UE m includes the local processing latency, wireless transmission latency, and edge processing latency. Based on the above discussion and analysis, the overall DT latency can be expressed as follows

$$T_m^{\mathsf{e2e}}(\mathbf{s}_m) = T_m^{\mathsf{ue}} + T_m^{\mathsf{co}} + T_m^{\mathsf{es}} = \frac{\alpha_m C_m}{f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}} + \max_{\forall k \in \mathcal{K}} \left\{ \frac{\beta_{mk} D_m}{R_{mk}(b_{mk}, p_{mk})} \right\} + \max_{\forall k \in \mathcal{K}} \left\{ \frac{\beta_{mk} C_m}{f_{mk}^{\mathsf{es}} - \hat{f}_{mk}^{\mathsf{es}}} \right\} \quad (16)$$

where $\mathbf{s}_m = \{\alpha_m, \beta_{mk}, f_k^{\mathsf{ue}}, f_{mk}^{\mathsf{es}}, b_{mk}, p_{mk}\}_{\forall m,k}$. Since each UE can offload the computational tasks to multiple ESs simultaneously, the max operator (i.e. $\max\{.\}_{\forall k}$) is applied for the transmission latency and edge processing latency.

2) Energy Consumption Model: The total energy consumption of UE k includes the energy consumed for computation (E_m^{cp}) and communication (E_m^{cm}), which is modelled as

$$E_m^{\mathsf{tot}}(\alpha_m, f_m^{\mathsf{ue}}, b_{mk}, p_{mk}) = E_m^{\mathsf{cp}} + E_m^{\mathsf{cm}} = \alpha_m \frac{\theta_m}{2} C_m \big(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^2 + \sum_{k \in \mathcal{K}} \frac{\beta_{mk} p_{mk} D_m}{R_{mk}(b_{mk}, p_{mk})} \quad (17)$$

where θ_m is the effective capacitance coefficient depending on the chipset of UE m [28].

3) FALM Problem Formulation: In this paper, we aim at minimising the summed e2e DT latency of all UEs which can be expressed as $T_\Sigma(\{\mathbf{s}_m\}) \triangleq \sum_{\forall m \in \mathcal{M}} T_m^{\mathsf{e2e}}(\mathbf{s}_m)$.

Definition 1 (Fairness of latency distribution)**.** *The optimal solution $\{\mathbf{s}_m^*\}$ is said to provide a fairer solution to the objective function $T_\Sigma(\{\mathbf{s}_m^*\})$ than the solution $\{\mathbf{s}_m\}$ if and only if $T_\Sigma(\{\mathbf{s}_m^*\})$ offers more uniform latency among all UEs than $T_\Sigma(\{\mathbf{s}_m\})$.*

Following Definition 1, we introduce new constant parameters $c_m \triangleq C_m / \sum_{m \in \mathcal{M}} C_m$ and $q > 0$ to re-weight the objective function $T_\Sigma(\{\mathbf{s}_m\})$ as follows:

$$T_\Sigma^q(\{\mathbf{s}_m\}) \triangleq \sum_{m \in \mathcal{M}} c_m \frac{T_m^{\mathsf{e2e}}(\mathbf{s}_m)^{q+1}}{q+1} \quad (18)$$

which is inspired from α-fairness framework [37]. The introduction of c_m can naturally improve the latency fairness among UEs by further optimising UEs which require more computation resources. We note that the positive parameter q is imposed to adjust the level of fair resource allocation. There is a trade off between the users fairness and computation complexity with respect to the increasing value of q parameter. In addition, if the parameter q is sufficiently large ($q \to \infty$), it becomes a min-max optimisation problem, i.e. $\min\limits_{\{\mathbf{s}_m\}} \max\limits_{\forall m} \{T_m^{\mathsf{e2e}}(\mathbf{s}_m)\}$.

The goal of the FALM problem is to find an optimal resource allocation strategy over $\mathbf{s} \triangleq \{\mathbf{s}_m\}_{\forall m}$ to minimise the sum of the individual e2e latency, which is mathematically formulated as

$$\min_{\mathbf{s}} \quad T_\Sigma^q(\{\mathbf{s}_m\}) \triangleq \sum_{m \in \mathcal{M}} c_m \frac{T_m^{\mathsf{e2e}}(\mathbf{s}_m)^{q+1}}{q+1} \quad (19a)$$

$$\text{s.t. } T_m^{\mathsf{e2e}}(\mathbf{s}_m) \le T_m^{\max}, \ \forall m \quad (19b)$$

$$\alpha_m + \sum_{k \in \mathcal{K}} \beta_{mk} = 1, \forall m \quad (19c)$$

$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} b_{mk} \le 1 \quad (19d)$$

$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} \beta_{mk} f_{mk}^{\mathsf{es}} \le F_{\max}^{\mathsf{es}} \quad (19e)$$

$$R_{mk}(b_{mk}, p_{mk}) \ge R_{\min}, \ \forall m, k \quad (19f)$$

$$E_m^{\mathsf{tot}}(\alpha_m, f_m^{\mathsf{ue}}, b_{mk}, p_{mk}) \le E_m^{\max}, \ \forall m \quad (19g)$$

$$\sum_{k \in \mathcal{K}} p_{mk} \le P_m^{\max}, \ \forall m \quad (19h)$$

$$\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathscr{D}, \mathbf{p} \in \mathscr{P}, \mathbf{f} \in \mathscr{F}, \mathbf{b} \in \mathscr{B} \quad (19i)$$

where $\mathscr{D} \triangleq \{\alpha_m, \beta_{mk}, \forall m, k | 0 \le \alpha_m \le 1, 0 \le \beta_{mk} \le 1, \forall m, k\}$, $\mathscr{P} \triangleq \{p_{mk}, \forall m, k | 0 \le p_{mk} \le P_m^{\max}, \forall m\}$, $\mathscr{F} \triangleq \{f_m^{\mathsf{ue}}, f_{mk}^{\mathsf{es}} \ \forall m, k | 0 \le f_m^{\mathsf{ue}} \le F_m^{\max}, \forall m, 0 \le f_{mk}^{\mathsf{es}} \le F_k^{\max}, \forall m, k\}$, and $\mathscr{B} \triangleq \{b_{mk}, \forall m, k | 0 \le b_{mk} \le 1\}$ are the set constraints of offloading decisions, uplink transmission power, processing rates and bandwidth coefficient, respectively. Constraint (19b) is the maximum latency budget for every incoming task, while (19c) is the constraint of offloading factors. Constraint (19e) ensures the required computation resource of ESs does not exceed the maximum capacity. Constraints (19f) and (19g) are the minimum transmission rate requirement for uplink transmission and the maximum energy consumption requirement of UEs, respectively.

III. Centralised Solution for the FALM Problem (19)

The problem (19) is highly complicated due to the non-convexity of the objective function (19a) and non-convex constraints (19b), (19e), (19f) and (19g). In addition, the objective function (19a) is non-smooth in $\mathbf{s}$ due to the $\max(,)$ operator expressed in (16), making it impossible to solve directly by the SCA method. To tackle this issue, we introduce new variables $\boldsymbol{\tau} \triangleq \{\tau_m\}_{\forall m}$ to rewrite (19) equivalently as:

$$\min_{\mathbf{s}, \boldsymbol{\tau}} \quad T_\Sigma^q(\boldsymbol{\tau}) \triangleq \sum_{m \in \mathcal{M}} c_m \frac{\tau_m^{q+1}}{q+1} \quad (20a)$$

$$\text{s.t. } \tau_m \ge T_m^{\mathsf{e2e}}(\mathbf{s}_m), \ \forall m \quad (20b)$$

$$\tau_m \le T_m^{\max}, \ \forall m \quad (20c)$$

$$(19c), (19d), (19e), (19f), (19g), (19h), (19i) \quad (20d)$$

where constraint (19b) is transformed to (20c). We are now in position to apply SCA to convexify the nonconvex parts of (20) and then develop an iterative method to solve the approximate convex program in a centralised fashion.

A. Approximate Convex Problem

Convexity of constraint (19e): By the principles of SCA [38], [39], we apply the following inequality:

$$xy \le \frac{1}{2} \left(\frac{\bar{y}}{\bar{x}} x^2 + \frac{\bar{x}}{\bar{y}} y^2 \right) \quad (21)$$

for $x > 0$ and $y > 0$. At the i-th iteration of the proposed iterative algorithm, we define $x = \beta_{mk}$, $\bar{x} = \beta_{mk}^{(i)}$, $y = f_{mk}^{\mathsf{es}}$,

$\bar{y} = f_{mk}^{\mathsf{es},(i)}$ to approximate (19e) as follows

$$\psi^{(i)}(\beta_{mk}, f_{mk}^{\mathsf{es}}) \triangleq \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} \frac{1}{2}\left(\frac{f_{mk}^{\mathsf{es},(i)}\beta_{mk}^2}{\beta_{mk}^{(i)}} + \frac{\beta_{mk}^{(i)}(f_{mk}^{\mathsf{es}})^2}{f_{mk}^{\mathsf{es},(i)}}\right) \leq F_{\max}^{\mathsf{es}} \tag{22}$$

where $\beta_{mk}^{(i)}$ and $f_{mk}^{\mathsf{es},(i)}$ are constant and feasible points of β_{mk} and f_{mk}^{es}, respectively. We note that $\psi^{(i)}(\beta_{mk}, f_{mk}^{\mathsf{es}}) = \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\beta_{mk}f_{mk}^{\mathsf{es}}$ whenever $\beta_{mk}^{(i)} = \beta_{mk}$ and $f_{mk}^{\mathsf{es},(i)} = f_{mk}^{\mathsf{es}}$.

Convexity of constraint (19f): The rate function $R_{mk}(b_{mk}, p_{mk})$ is quite complicated and a direct application of SCA is inapplicable. Following [8] and [12], we can see that $V_{mk} \approx 1$ when $\gamma_{mk}(p_{mk}) \geq \bar{\gamma} = 5$ dB, i.e.

$$p_{mk} \geq \frac{\bar{\gamma}}{(L-1)\sigma_{mk}^2 - \bar{\gamma}\delta_{mk}^2}. \tag{23}$$

Under the condition (23), we rewrite $R_{mk}(b_{mk}, p_{mk})$ as

$$R_{mk}(b_{mk}, p_{mk}) \triangleq \frac{(1-\omega_k)B}{\ln 2}\Big[G_{mk}(b_{mk}, p_{mk}) - W_{mk}(b_{mk})\Big] \tag{24}$$

where $G_{mk}(b_{mk}, p_{mk}) = b_{mk}\ln(1+\gamma_{mk}(p_{mk}))$ and $W_{mk}(b_{mk}) = \sqrt{b_{mk}}\frac{Q^{-1}(\epsilon_{mk})}{\sqrt{\phi B}}$. Following the derivations given in the Appendix, the transmission rate $R_{mk}(b_{mk}, p_{mk})$ can be innerly approximated around the feasible point $(b_{mk}^{(i)}, p_{mk}^{(i)})$ as

$$\begin{aligned} R_{mk}(b_{mk}, p_{mk}) &\geq \frac{(1-\omega_k)B}{\ln 2}\Big[\mathcal{G}_{mk}^{(i)}(b_{mk}, p_{mk}) - \mathcal{W}_{mk}^{(i)}(b_{mk})\Big] \\ &\triangleq R_{mk}^{(i)}(b_{mk}, p_{mk}) \end{aligned} \tag{25}$$

where $\mathcal{G}_{mk}^{(i)}(b_{mk}, p_{mk})$ and $\mathcal{W}_{mk}^{(i)}(b_{mk})$ are defined as in (54) and (56) in the Appendix, respectively. As a result, we can iteratively replace (19f) by the following convex constraint

$$R_{mk}^{(i)}(b_{mk}, p_{mk}) \geq R_{\min}, \forall m, k. \tag{26}$$

Convexity of constraint (19g): Given the concave function $R_{mk}^{(i)}(b_{mk}, p_{mk})$ in (25), we first introduce new variables $\mathbf{r} \triangleq \{r_{mk}\}_{\forall m,k}$ to express (19g) equivalently as

$$\begin{cases} \dfrac{\theta}{2}C_m\alpha_m\big(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^2 + \displaystyle\sum_{k\in\mathcal{K}} D_m\beta_{mk}p_{mk}r_{mk} \leq E_m^{\max}, \forall m & \text{(27a)} \\ \dfrac{1}{R_{mk}^{(i)}} \leq r_{mk}, \forall m, k & \text{(27b)} \end{cases}$$

where constraint (27b) is now convex. By applying (21) for the first part of the left-hand side (LHS) of (27a) with $x = \alpha_m$, $\bar{x} = \alpha_m^{(i)}$, $y = \big(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^2$, and $\bar{y} = \big(f_m^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}}\big)^2$ and introducing new variables $\boldsymbol{\varphi} \triangleq \{\varphi_{mk}\}_{\forall m,k}$, we can equivalently approximate (27a) as follows

$$\begin{cases} \dfrac{\theta C_m}{4}\left(\dfrac{\big(f_m^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}}\big)^2}{\alpha_m^{(i)}}\alpha_m^2 + \dfrac{\alpha_m^{(i)}\big(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^4}{\big(f_m^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}}\big)^2}\right) + \displaystyle\sum_{k\in\mathcal{K}} D_m\varphi_{mk}^2 \leq E_m^{\max}, \forall m & \text{(28a)} \\ p_{mk}r_{mk} \leq \dfrac{\varphi_{mk}^2}{\beta_{mk}}, \forall m, k & \text{(28b)} \end{cases}$$

where constraint (28a) is now convex. Next, by using (21) to find a convex upper bound of $p_{mk}r_{mk}$ and the following inequality [40]

$$\frac{x^2}{y} \geq \frac{2\bar{x}}{\bar{y}}x - \frac{\bar{x}^2}{\bar{y}^2}y \tag{29}$$

to linearise $\frac{\varphi_{mk}^2}{\beta_{mk}}$, the nonconvex constraint (28b) can be approximated as

$$\frac{1}{2}\Big(\frac{p_{mk}^{(i)}}{r_{mk}^{(i)}}r_{mk}^2 + \frac{r_{mk}^{(i)}}{p_{mk}^{(i)}}p_{mk}^2\Big) \leq \frac{2\varphi_{mk}^{(i)}\varphi_{mk}}{\beta_{mk}^{(i)}} - \frac{\varphi_{mk}^{2(i)}\beta_{mk}}{\big(\beta_{mk}^{(i)}\big)^2}, \ \forall m,k. \tag{30}$$

Convexity of (20b): We first find the concave upper bound of the e2e latency $T_m^{\mathsf{e2e}}(\mathbf{s}_m)$. By using $\mathbf{r}$ defined as in (27b), it follows that

$$T_m^{\mathsf{e2e}}(\mathbf{s}_m) \leq \frac{\alpha_m C_m}{f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}} + \max_{\forall k\in\mathcal{K}}\{D_m\beta_{mk}r_{mk}\} + \max_{\forall k\in\mathcal{K}}\left\{\frac{\beta_{mk}C_m}{f_{mk}^{\mathsf{es}} - \hat{f}_{mk}^{\mathsf{es}}}\right\} \tag{31}$$

which is innerly convexified as

$$\begin{aligned} T_m^{\mathsf{e2e}}(\mathbf{s}_m) &\leq \tau^{\mathsf{ue},(i)}(\alpha_m, f_m^{\mathsf{ue}}) + \tau^{\mathsf{co},(i)}(\beta_{mk}, r_{mk}) \\ &\quad + \tau^{\mathsf{es},(i)}(\beta_{mk}, f_{mk}^{\mathsf{es}}) \triangleq \mathcal{T}_m^{\mathsf{e2e},(i)}(\mathbf{s}_m) \end{aligned} \tag{32}$$

where

$$\tau^{\mathsf{ue},(i)}(\alpha_m, f_m^{\mathsf{ue}}) \triangleq \frac{C_m}{2}\Big(\frac{\alpha_m^{(i)}\big(f_m^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}}\big)}{\big(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^2} + \frac{1}{f_m^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}}}\frac{\alpha_m^2}{\alpha_m^{(i)}}\Big)$$

$$\tau^{\mathsf{co},(i)}(\beta_{mk}, r_{mk}) \triangleq \max\Big\{\frac{D_m}{2}\Big(\frac{\beta_{mk}^{(i)}}{r_{mk}^{(i)}}r_{mk}^2 + \frac{r_{mk}^{(i)}}{\beta_{mk}^{(i)}}\beta_{mk}^2\Big)\Big\}$$

$$\tau^{\mathsf{es},(i)} \triangleq \max\Big\{\frac{C_m}{2}\Big(\frac{\beta_{mk}^{(i)}\big(f_{mk}^{\mathsf{es},(i)} - \hat{f}_{mk}^{\mathsf{es}}\big)}{\big(f_{mk}^{\mathsf{es}} - \hat{f}_{mk}^{\mathsf{es}}\big)^2} + \frac{1}{f_{mk}^{\mathsf{es},(i)} - \hat{f}_{mk}^{\mathsf{es}}}\frac{\beta_{mk}^2}{\beta_{mk}^{(i)}}\Big)\Big\}.$$

As a result, constraints (20b) is iteratively replaced by

$$\tau_m \geq \mathcal{T}_m^{\mathsf{e2e},(i)}(\mathbf{s}_m), \forall m. \tag{33}$$

Summing up, the SCA-based approximate convex program of problem (20) solved at the i-th iteration is given as

$$\min_{\mathbf{s},\boldsymbol{\tau},\mathbf{r},\boldsymbol{\varphi}} \quad \sum_{m\in\mathcal{M}} c_m\frac{\tau_m^{q+1}}{q+1} \tag{34a}$$

$$\text{s.t. } (19c), (19d), (19h), (19i), (22), (23), (26), (27b), (28a), (30), (33). \tag{34b}$$

B. Proposed Centralised Algorithm

The SCA-based algorithm for solving the FALM problem (19) is summarised in Algorithm 1. We successively solve the approximate convex program (34) in Step 2 to obtain the optimal solution $(\mathbf{s}^*, \boldsymbol{\tau}^*, \mathbf{r}^*, \boldsymbol{\varphi}^*)$, which is then updated in Step 3. This procedure is repeated until convergence or the maximum number of iterations is reached. Towards an efficient algorithm, we first generate an initial feasible point for problem (34). In particular, the offloading variables for local execution at UEs are set to $\alpha_m = 0.5, \forall m, \beta_{mk} = 0.5/K, \forall m, k$. The

Algorithm 1 Proposed SCA-based Centralised Algorithm for Solving the FALM Problem (19)

Initialisation: Set $i = 0$ and generate an initial feasible point $(\mathbf{s}^{(0)}, \boldsymbol{\tau}^{(0)}, \mathbf{r}^{(0)}, \boldsymbol{\varphi}^{(0)})$ satisfying constraints in (34); Set the tolerance $\varepsilon = 10^{-3}$ and the maximum number of iterations $I^{\max} = 10$.

1: **repeat**
2: Solve problem (34) for given $(\mathbf{s}^{(i)}, \boldsymbol{\tau}^{(i)}, \mathbf{r}^{(i)}, \boldsymbol{\varphi}^{(i)})$ to obtain the optimal solution denoted by $(\mathbf{s}^*, \boldsymbol{\tau}^*, \mathbf{r}^*, \boldsymbol{\varphi}^*)$;
3: Update $(\mathbf{s}^{(i+1)}, \boldsymbol{\tau}^{(i+1)}, \mathbf{r}^{(i+1)}, \boldsymbol{\varphi}^{(i+1)}) := (\mathbf{s}^*, \boldsymbol{\tau}^*, \mathbf{r}^*, \boldsymbol{\varphi}^*)$;
4: Set $i := i + 1$;
5: **until** Convergence or $i > I^{\max}$
6: **Output**: $(\mathbf{s}^*, \boldsymbol{\tau}^*, \mathbf{r}^*, \boldsymbol{\varphi}^*)$

processing rate of UEs is initiated with $f_m^{\mathsf{ue}} = F_{\min}^{\mathsf{ue}}, \forall m$, while other resource variables of UEs and ESs are equally set with respect to the resource budgets and latency requirements. Nevertheless, Algorithm 1 is performed at a central node that requires to collect information of the entire network and does not fully exploit the computation power of ESs. Therefore, we further investigate a distributed optimisation solution for this problem.

IV. Distributed Solution for Minimising The Total e2e Latency

A. Distributed Problem Transformation

In order to develop a distributed approach for solving (20), we first transform it into the separated problems which can be solved in parallel with K ESs based on their local information. The most critical challenge is that there are several constraints, where the variables are mutual dependence with respect to the global budget of the system, i.e. the bandwidth allocation constraint (19d) and ESs' computing capacity (19e). To separate the centralised problem into K subproblems, we categorise the considered variables into two types based on their mutual dependence among K ESs, including *independent variables* and *dependent variables*. The offloading portion variables $(\alpha_m, \forall m)$ and the processing rate of UE $(f_m^{\mathsf{ue}}, \forall m)$ are independent which can be separated into K variables. It is clear that the other variables (e.g. bandwidth allocation, transmission power and the processing rate of ESs) are dependent variables, which require additional operations at the global level to guarantee the optimality.

Let us define $\mathbf{x}_k = \{\alpha_{mk}, \beta_{mk}, p_{mk}, b_{mk}, f_{mk}^{\mathsf{ue}}, f_{mk}^{\mathsf{es}}\}_{\forall m \in \mathcal{M}}$ as the set of the local optimisation variables stored at ES k. For the separable variables $(\boldsymbol{\alpha}, \mathbf{f}^{\mathsf{ue}})$, we introduce new variables $z_m^{(\alpha)}$ and $z_m^{(f)}$ to represent as the global versions of the local variables α_{mk} and f_{mk}^{ue} of task J_m. To separate β_{mk} and p_{mk}, we introduce new global variables $z_{mk}^{(\beta)}$ and $z_{mk}^{(p)}$ to re-express constraints (44c) and (19h) as

$$\alpha_{mk} + \beta_{mk} + z_{mk}^{(\beta)} = 1, \ \forall m \tag{35a}$$

$$p_{mk} + z_{mk}^{(p)} \leq P_m^{\max}, \ \forall m \tag{35b}$$

where $z_{mk}^{(\beta)}$ and $z_{mk}^{(p)}$ are globally updated as

$$z_{mk}^{(\beta)} = \sum_{k' \neq k}^{K} \beta_{mk'}, \ \forall m, k \tag{36a}$$

$$z_{mk}^{(p)} = \sum_{k' \neq k}^{K} p_{mk'}, \ \forall m, k. \tag{36b}$$

Next, we introduce a new global variable $z_k^{(b)}$ to rewrite the constraint (19d) as follows

$$\sum_{m=1}^{M} b_{mk} + z_k^{(b)} \leq 1, \forall k \tag{37}$$

where $z_k^{(b)}$ is globally updated as

$$z_k^{(b)} = \sum_{m=1}^{M} \sum_{k' \neq K}^{K} b_{mk'}, \ \forall m, k. \tag{38}$$

Similarly, let us define

$$z_k^{\mathsf{es}} = \sum_{m=1}^{M} \sum_{k' \neq k}^{K} \beta_{mk'} f_{mk'}^{\mathsf{es}} \tag{39}$$

to rewrite constraint (19e) as

$$\sum_{m=1}^{M} \frac{1}{2} \Big(\frac{f_{mk}^{\mathsf{es}(i)}}{\beta_{mk}^{(i)}} (\beta_{mk})^2 + \frac{\beta_{mk}^{(i)}}{f_{mk}^{\mathsf{es}(i)}} (f_{mk}^{\mathsf{es}})^2 \Big) + z_k^{\mathsf{es}} \leq F_{\max}^{\mathsf{es}}, \forall k \tag{40}$$

which can be solved locally at ES k. Moreover, constraint (19g) can be locally separated ES k as

$$E_m^{\mathrm{cp},(i)} + E_{mk}^{\mathrm{cm},(i)} + \sum_{k' \neq k}^{K} E_{mk'}^{\mathrm{cm},(i)} \leq E_m^{\max}, \ \forall m \tag{41}$$

where $E_{mk}^{\mathrm{cm},(i)}$ and $E_{mk'}^{\mathrm{cm},(i)}$ are the energy consumption for uplink transmission from UE m to ES k and ES $k' \neq k$, respectively, which are defined as follows

$$E_{mk}^{\mathrm{cm},(i)} = D_m \varphi_{mk}^2, \ \forall m \tag{42a}$$

$$E_{mk'}^{\mathrm{cm},(i)} = D_m \varphi_{mk'}^2, \ \forall m, k' \tag{42b}$$

with r_{mk} and φ_{mk} begin defined in (27b) and (28b).

Finally, based on the development of the centralised solution in Section III, the summed e2e DT latency stored at ES k is given as

$$t_k(\mathbf{x}_k) \triangleq \sum_{m \in \mathcal{M}} c_m \frac{\big(\mathcal{T}_{mk}^{(i)}(\mathbf{x}_k)\big)^{q+1}}{q+1}, \ \forall k \tag{43}$$

where

$$\mathcal{T}_{mk}^{(i)}(\mathbf{x}_k) \triangleq \frac{1}{2} D_m \Big(\frac{\beta_{mk}^{(i)}}{r_{mk}^{(i)}} r_{mk}^2 + \frac{r_{mk}^{(i)}}{\beta_{mk}^{(i)}} \beta_{mk}^2 \Big)$$
$$+ \frac{C_m}{2} \Big[\frac{\alpha_{mk}^{(i)} (f_{mk}^{\mathsf{ue},(i)} - \hat{f}_m^{\mathsf{ue}})}{\big(f_{mk}^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}\big)^2} + \frac{1}{f_{mk}^{\mathsf{ue},(i)} - \hat{f}_{mk}^{\mathsf{ue}}} \frac{\alpha_{mk}^2}{\alpha_{mk}^{(i)}} \Big]$$
$$+ \frac{C_m}{2} \Big[\frac{\beta_{mk}^{(i)} (f_{mk}^{\mathsf{es},(i)} - \hat{f}_{mk}^{\mathsf{es}})}{\big(f_{mk}^{\mathsf{es}} - \hat{f}_{mk}^{\mathsf{es}}\big)^2} + \frac{1}{f_{mk}^{\mathsf{es},(i)} - \hat{f}_{mk}^{\mathsf{es}}} \frac{\beta_{mk}^2}{\beta_{mk}^{(i)}} \Big] \ \forall m, k.$$

Keeping the above discussion in mind, problem (19) can be equivalently transformed into the following distributed convex

problem:

$$\underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \ \sum_{k\in\mathcal{K}} t_k(\mathbf{x}_k) \tag{44a}$$

$$\text{s.t. } \mathcal{T}_{mk}^{(i)}(\mathbf{x}_k) \le T_m^{\max}, \forall m,k \tag{44b}$$

$$\boldsymbol{\alpha}_k = \mathbf{z}^{(\alpha)}, \forall k \tag{44c}$$

$$\mathbf{f}_k^{\mathsf{ue}} = \mathbf{z}^{(f)}, \forall k \tag{44d}$$

$$(19f), (19i), (23), (35a), (35b), (37), (40), (41) \tag{44e}$$

where $\mathbf{x} \triangleq \{\mathbf{x}_k\}_{\forall k}$ and $\mathbf{z} \triangleq \{\mathbf{z}^{(\alpha)}, \mathbf{z}^{(\beta)}, \mathbf{z}^{(f)}, \mathbf{z}^{(b)}, \mathbf{z}^{(p)}, \mathbf{z}^{\mathsf{es}}\}$ represent the local and global variables, respectively.

To solve problem (44) at ES $k \in \mathcal{K}$, let us define $\boldsymbol{\alpha}_k \triangleq \{\alpha_{mk}\}_{\forall m}$, $\mathbf{f}_k^{\mathsf{ue}} \triangleq \{f_{mk}^{\mathsf{ue}}\}_{\forall m}$, $\mathbf{z}^{(\alpha)} \triangleq \{z_m^{(\alpha)}\}_{\forall m}$, $\mathbf{z}^{(\beta)} \triangleq \{z_{mk}^{(\beta)}\}_{\forall m,k}$, $\mathbf{z}^{(f)} \triangleq \{z_m^{(f)}\}_{\forall m}$, $\mathbf{z}^{(b)} \triangleq \{z_k^{(b)}\}_{\forall k}$, $\mathbf{z}^{(p)} \triangleq \{z_{mk}^{(p)}\}_{\forall m,k}$, and $\mathbf{z}^{\mathsf{es}} \triangleq \{z_k^{\mathsf{es}}\}_{\forall k}$. The proposed ADMM-based algorithm to solve (44) in a distributed fashion will be deatiled next.

B. Proposed ADMM-based Consensus Optimisation Solution

The augmented Lagrangian function of (44) is expressed as

$$L_\rho(\mathbf{x},\mathbf{z},\boldsymbol{\psi},\boldsymbol{\xi}) = \sum_{k\in\mathcal{K}} \Big[t_k(\mathbf{x}_k) + \boldsymbol{\psi}_k^T(\boldsymbol{\alpha}_k - \mathbf{z}^{(\alpha)}) + \boldsymbol{\xi}_k^T(\mathbf{f}_k^{\mathsf{ue}} - \mathbf{z}^f) + \frac{\rho}{2}\left(\|\boldsymbol{\alpha}_k - \mathbf{z}^{(\alpha)}\|_2^2 + \|\mathbf{f}_k^{\mathsf{ue}} - \mathbf{z}^f\|_2^2\right)\Big] \tag{45}$$

where $\rho > 0$ is a penalty parameter, and $\boldsymbol{\psi} \triangleq \{\boldsymbol{\psi}_k^T\}_{\forall k}$ and $\boldsymbol{\xi} \triangleq \{\boldsymbol{\xi}_k^T\}_{\forall k}$ are the Lagrange multipliers associated with (44c) and (44d), respectively. We note that the panalty function $\frac{\rho}{2}\left(\|\boldsymbol{\alpha}_k - \mathbf{z}^{(\alpha)}\|_2^2 + \|\mathbf{f}_k^{\mathsf{ue}} - \mathbf{z}^f\|_2^2\right)$ is strictly convex, leading to the convexity of (45). By ADMM principles [41], the local variables, global variables and Lagrange multipliers need to be updated in each iteration of the SCA approach.

Update of the local variables: We denote by $\mathscr{C}_k^{(i)}$ the feasible set of ES k at the i-th iteration of SCA, satisfying

$$\mathscr{C}_k^{(i)} \triangleq \{\mathbf{x}_k | \text{s.t. } (44b), (44e)\} \tag{46}$$

where $\mathbf{x}_k$ is the set of the local variables. To update the local variables, we solve the following convex problem

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}_k\in\mathscr{C}_k^{(i)},\forall k\in\mathcal{K}}{\operatorname{argmin}} \ L_\rho(\mathbf{x},\mathbf{z}^{(i)},\boldsymbol{\psi}^{(i)},\boldsymbol{\xi}^{(i)}). \tag{47}$$

It is clear that problem (47) can be decomposed into K subproblems which are solved locally at ESs. Given the local variables and updates from the exchanged information from other ESs via global updates, the k-th ES solves its local problem as

$$\mathbf{x}_k^{(i+1)} = \underset{\mathbf{x}_k\in\mathscr{C}_k^{(i)}}{\operatorname{argmin}} \ t_k(\mathbf{x}_k) + \boldsymbol{\psi}_k^{(i)T}(\boldsymbol{\alpha}_k - \mathbf{z}^{\alpha(i)}) + \boldsymbol{\xi}_k^{(i)T}(\mathbf{f}_k^{\mathsf{ue}} - \mathbf{z}^{f(i)}) + \frac{\rho}{2}\left(\|\boldsymbol{\alpha}_k - \mathbf{z}^{\alpha(i)}\|_2^2 + \|\mathbf{f}_k^{\mathsf{ue}} - \mathbf{z}^{f(i)}\|_2^2\right). \tag{48}$$

Update of the global variables: We first note that the global variables $(\mathbf{z}^{(\beta)}, \mathbf{z}^p, \mathbf{z}^{(b)}, \mathbf{z}^{\mathsf{es}})$ are updated by following (36a), (36b), (38) and (39). The rest of the global variables $(\mathbf{z}^{(\alpha)}, \mathbf{z}^{(f)})$ is found by solving the following problem:

$$\mathbf{z}^{(i+1)} = \underset{\mathbf{z}^{(\alpha)},\mathbf{z}^{(f)}}{\operatorname{argmin}} \ L_\rho(\mathbf{x}^{(i+1)},\mathbf{z},\boldsymbol{\psi}^{(i)},\boldsymbol{\xi}^{(i)}). \tag{49}$$

Update of the Lagrange multipliers: Given the updated global variables $\mathbf{z}^{(i+1)}$ and the updated local variables $\mathbf{x}_k^{(i+1)}$, the Lagrange multipliers $\boldsymbol{\psi}_k$ and $\boldsymbol{\xi}_k$ are updated as follows

$$\boldsymbol{\psi}_k^{(i+1)} = \boldsymbol{\psi}_k^{(i)} + \rho\left(\boldsymbol{\alpha}_k^{(i+1)} - \mathbf{z}^{\alpha(i+1)}\right) \tag{50a}$$

$$\boldsymbol{\xi}_k^{(i+1)} = \boldsymbol{\xi}_k^{(i)} + \rho\left(\mathbf{f}_k^{\mathsf{ue}(i+1)} - \mathbf{z}^{f(i+1)}\right) \tag{50b}$$

which do not require additional exchange of information among ESs.

C. Proposed Consensus Algorithm for Distributed Solution

Algorithm 2 Proposed ADMM-based Consensus Algorithm for Solving (44)

1: **Initialisation**: Set $i = 0$, generate the initial feasible points $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}, \mathbf{p}^{(0)}, \mathbf{b}^{(0)}, \mathbf{f}^{(0)})$, and choose the initial values for $\mathbf{z}^{(0)}$, $(\boldsymbol{\psi}^{(0)}, \boldsymbol{\xi}^{(0)})$.
2: **while** *(not convergent)* **do**
3: *Local updates*:
4: **for** $k \in \mathcal{K}$ *in parallel* **do**
5: ES k updates local variables $\mathbf{x}_k^{(i+1)}$ by solving the convex problem (48)
6: Exchange the local values for global consensus
7: **end for**
8: *Global updates*:
9: Update the global variables $(\mathbf{z}^{\alpha,(i+1)}, \mathbf{z}^{\beta,(i+1)})$ by (49), and $\left(\mathbf{z}^{\beta,(i+1)}, \mathbf{z}^{p,(i+1)}, \mathbf{z}^{b,(i+1)}, \mathbf{z}^{\mathsf{es},(i+1)}\right)$ by following (36a), (36b), (38) and (39)
10: Update the Lagrange multipliers $\boldsymbol{\psi}_k^{(i+1)}, \boldsymbol{\xi}_k^{(i+1)}$
11: Update the SCA parameters $\mathbf{s}^{(i+1)} = \mathbf{s}^*$
12: Set $i := i + 1$
13: **end while**

Based on the above development, the ADMM-based consensus algorithm for solving (44) is summarised in Algorithm 2. We consider that there exists an information technology infrastructure to allow all necessary information being exchanged among K ESs before performing the global updates (i.e. Step 6). In the current cellular communication systems, APs or gNB uses the X2 interface to intercommunicate with each other directly. An exemplary illustration of the distributed solution with two ESs is provided in Fig. 2. In this diagram, each ES first solves its own problem to obtain the optimal solutions of the local variables, and then, the global updates are performed to find the next values of global variables. Finally, the multiplier parameters are updated for the next iteration. The procedure is repeated until convergence.

V. Complexity and Convergence Analysis of The Proposed Solutions

A. Complexity Analysis

1) Complexity of the Centralised Algorithm: The approximate convex problem (34) consists of $6MK + 3M$ scalar decision variables and $7MK + 7M + 2$ linear or quadratic constraints. Therefore, the per-iteration computational complexity of Algorithm 1 for solving (34) is thus $\mathcal{O}\big(\sqrt{7MK + 7M + 2}(6MK + 3M)^2\big)$ [42, Sec. 6]. It is clear that as the number of UEs increases, the computational complexity of the centralised approach increases significantly, resulting in higher execution times.

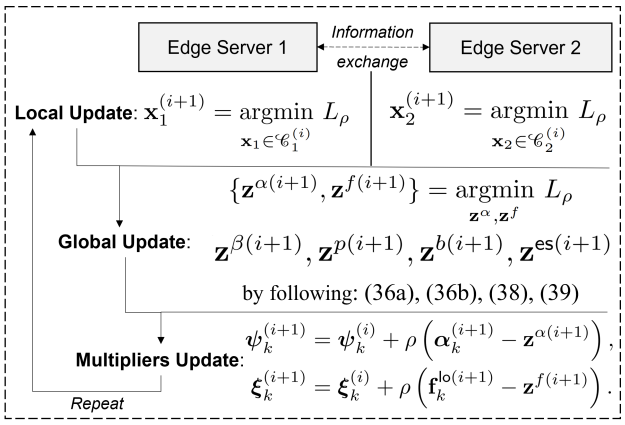


Fig. 2: An exemplary illustration of the distributed solution with two ESs.

2) Complexity of the Distributed Algorithm: In Algorithm 2, the subproblems are solved in parallel at K ESs to fully exploit their computing power. In addition, it also does not require a central server to collect the information and data of the whole network and carry out all the computation. In the local computation step, the major complexity comes from solving the convex program (48) under the feasible set (46). Problem (47) includes $7M$ scalar decision variables and $11M + 2$ constraints. The worst-case per-iteration computational complexity in the local update at each ES (i.e. Step 5 of Algorithm 2) is thus $\mathcal{O}\big(\sqrt{11M}(7M)^2\big)$, which is significantly lower than that in the centralised Algorithm 1.

B. Convergence Analysis

1) Convergence of the Centralised Algorithm: In Algorithm 1, the SCA principles are applied to approximate the nonconvex parts of the original problem (19). Algorithm 1 successively produces a sequence of improved points $\{\mathbf{s}^{(i)}, \boldsymbol{\tau}^{(i)}, \mathbf{r}^{(i)}, \boldsymbol{\varphi}^{(i)})\}$ and a sequence of non-increasing objective values. Let $\boldsymbol{\Psi}^* \triangleq \{\mathbf{s}^*, \boldsymbol{\tau}^*, \mathbf{r}^*, \boldsymbol{\varphi}^*)\}$ be a local minimiser of the objective function (34) denoted by $f(\boldsymbol{\Psi}^*)$. We can show that $f(\boldsymbol{\Psi}^{(i)}) > f(\boldsymbol{\Psi}^{(i+1)})$ with $\boldsymbol{\Psi}^{(i)} \neq \boldsymbol{\Psi}^{(i+1)}$ and $f(\boldsymbol{\Psi}^{(i)}) = f(\boldsymbol{\Psi}^{(i+1)})$ with $\boldsymbol{\Psi}^{(i)} = \boldsymbol{\Psi}^{(i+1)}$. According to [43, Sec. 7], the sequence of the objective values is bounded below due to the limited budgets of both computation and communication resources. For a sufficiently large number of iterations, Algorithm 1 is guaranteed to converge to at least a locally optimal solution of (20) (or the original problem (19)).

2) Convergence of the Distributed Algorithm: The convergence of an ADMM-based algorithm is already provided in [41, Sec. 3]. We first provide the following lemma to characterise the gap of the augmented Lagrangian function between two consecutive iterations.

Lemma 1. *The upper bound of the augmented Lagrangian function between two consecutive iterations in Algorithm 2 is given by*

$$\begin{aligned} &L_\rho^{(i+1)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi}) - L_\rho^{(i)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi}) \leq \\ &\sum_{k=1}^{K} \frac{1}{\rho}\Big(d^2 - \frac{\rho^2}{2}\Big)\Big(\|\boldsymbol{\psi}_k^{(i+1)} - \boldsymbol{\psi}_k^{(i)}\|_2^2 + \|\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{\xi}_k^{(i)}\|_2^2\Big) \\ &- \sum_{k=1}^{K} \frac{\rho}{2}\Big(\|\boldsymbol{\psi}_k^{(i+1)} - \boldsymbol{\psi}_k^{(i)}\|_2^2 + \|\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{\xi}_k^{(i)}\|_2^2\Big) \\ &\triangleq \Delta L_\rho^{(i)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi}) \end{aligned} \tag{51}$$

where ρ is the penalty constant introduced in (45) *and d is a positive constant depending on the specific optimisation problems.*

The proof of Lemma 1 is followed [41, Sec. 3]. Lemma 1 indicates an important insight that if ρ is chosen to be large enough, i.e. $\rho > d\sqrt{2}$, then the upper bound $L_\rho^{(i)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi})$ in (51) becomes negative. In other words, the sequence of the augmented Lagrangian values $\{L_\rho^{(i)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi})\}$ is non-increasing. We note that $L_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\xi})$ is bounded below by the limited budgets of both computation and communication resources, and thus guaranteeing the convergence of Algorithm 2.

C. Choice of Parameters for the Distributed Algorithm 2

In the ADMM-based algorithm, it is critical to choose appropriate parameters to obtain the best solution, as close to the centralized solution as possible. In particular, the choice of initial values of $(\boldsymbol{\psi}^{(0)}, \boldsymbol{\xi}^{(0)})$ has a strong impact on the convergence speed of optimising the global variables $(\mathbf{z}^{(\alpha)}, \mathbf{z}^{(f)})$. Since the difference between the value range of the offloading portions and the processing rates is considerably large (i.e. $[0, 1]$ and $[0, 1.5] \times 10^9$), the initial values of $(\boldsymbol{\psi}^{(0)}, \boldsymbol{\xi}^{(0)})$ need to be carefully adjusted to mitigate its negative impact. In addition, the choice of the positive fairness parameter q and the penalty parameter ρ of the augmented Lagrangian function also has a direct impact on the performance of Algorithm 2. For instance, a small value of the fairness parameter q cannot guarantee the latency fairness among UEs, while a large ρ may lead to an improper early termination. The impacts of these settings will be numerically elaborated in the next section.

VI. Simulation Results

This section validates the effectiveness of the proposed algorithms by conducting computer simulations. We first provide the simulation setup Section VI-A, and numerical results are then given in Section VI-B. All the simulation results are run in the MATLAB environment with CVX package. The convex problems are solved by **SDPT3** solver [44].

For performance comparison, we consider the following benchmark schemes:

- *Fixed bandwidth* [28]: The bandwidth allocation is not jointly optimised. The bandwidth portion allocated to each UE is obtained by equally dividing the whole system bandwidth with the number of UEs, i.e. $b_{mk} = 1/MK$.
- *Fixed processing rate* [12], [27]: The processing rates of UEs and ESs are fixed as in the initialisation step.
- *Fixed offloading* [28]: To demonstrate the effectiveness of offloading optimisation in the considerd DT-aided

TABLE I: Simulation Parameters

.95 Parameter	Value
Number of antennas, L	8 [9]
Maximum transmit power, $P_m^{\max}, \forall m$	23 dBm [30]
System bandwidth, B	10 MHz [21]
Transmission duration URLLC, ϕ	0.1 ms [30]
Decoding error probability, $\varepsilon_{mk}, \forall m, k$	10^{-7} [12]
Noise spectral density	-174 dBm/Hz [12]
Maximum UEs' processing rate, $F_{\max}^{\mathsf{ue}}$	1.5 GHz [12]
Total ES processing rate, $F_{\max}^{\mathsf{es}}$	10 GHz [21]
Input task size, $D_m, \forall m$	1354 bytes [5]
Task complexity, $\eta_m, \forall m$	$[100, 400]$ cycles/byte [12]
Total delay requirement, $T_m^{\max}, \forall m$	2 ms [21]
Minimum data rate, $R_{mk}^{\min}, \forall m, k$	2.57 Mbps
Max. UE's energy consump., $E_m^{\max}, \forall m$	2 mJ [12]
Computation power parameter, $\theta_m, \forall m$	10^{-26} Watt.s^3/cycle^3 [13], [45]

distributed edge computing, the offloading portions are set to be $\alpha_m = 0.5$, $\sum_{k\in\mathcal{K}} \beta_{mk} = 0.5, \forall m$.

The optimisation problems of these benchmark schemes can readily be formulated by slightly modifying the problem (19), and their solutions can be found by applying our proposed Algorithms 1 and 2.

A. Simulation Setup

In the DT-aided edge computing with URLLC, we consider the scenario with $M = 8$ UEs and $K = 2$ ESs, which are located within an area of 100 m $\times$ 100 m [45]. All UEs are randomly distributed while two ESs are located at the central positions of the area. The large-scale fading of the channel between the m-th UE and the k-th AP is modelled as $g_{mk} = 10^{\mathrm{PL}(d_{mk})/10}$, where $\mathrm{PL}(d_{mk}) = -35.3 - 37.6 \log_{10} d_{mk}$ denotes the pathloss in dB [9]; Herein, d_{mk} is the distance between the m-th UE and the k-th ES. The URLLC decoding error probability is set to 10^{-7}, and the noise spectral density is -174 dBm/Hz [9].

The maximum computing capacity of ESs is $F_{\max}^{\mathsf{es}} = 10$ GHz, while the maximum processing rate of UEs is $F_{\max}^{\mathsf{ue}} = 1.5$ GHz. The task complexity (i.e. $\eta_m \triangleq C_m/D_m, \forall m$) is uniformly distributed in the range of $[100, 400]$ cycles/byte [12]. Following the 3GPP Release 15 [5], the input data size is set to $D_m = 1354$ bytes, and the maximum delay requirement of each task is $T_m^{\max} = 2$ ms. Finally, the computation power parameter of energy consumption of UEs is set to $\theta_m = 10^{-26}$ Watt.s^3/cycle^3 [13], [45]. All simulation parameters are summarised in Table I.

B. Numerical Results and Discussions

1) Convergence behavior of Algorithms 1 and 2: To demonstrate the convergence behaviour of the proposed algorithms, we plot the total e2e latency as a function of the iteration index with $M = 8$ UEs, $K = 2$ ESs and different values of the fairness parameter $q = \{8, 10\}$. First, we can observe from Fig. 3 that the total e2e latency (i.e., the objective function) is monotonically decreasing and converges within about five iterations. More importantly, Algorithm 2 is likely to achieve the same performance as Algorithm 1, which confirms the effectiveness of the proposed distributed algorithm. Although the distributed solution requires more iterations to converge than the centralised method, its complexity of per-iteration and processing time in solving the local problems are significantly reduced. This is practically attractive for networks of medium-to-large sizes.

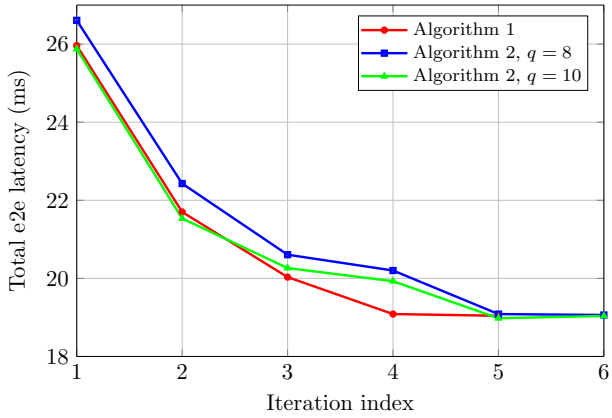
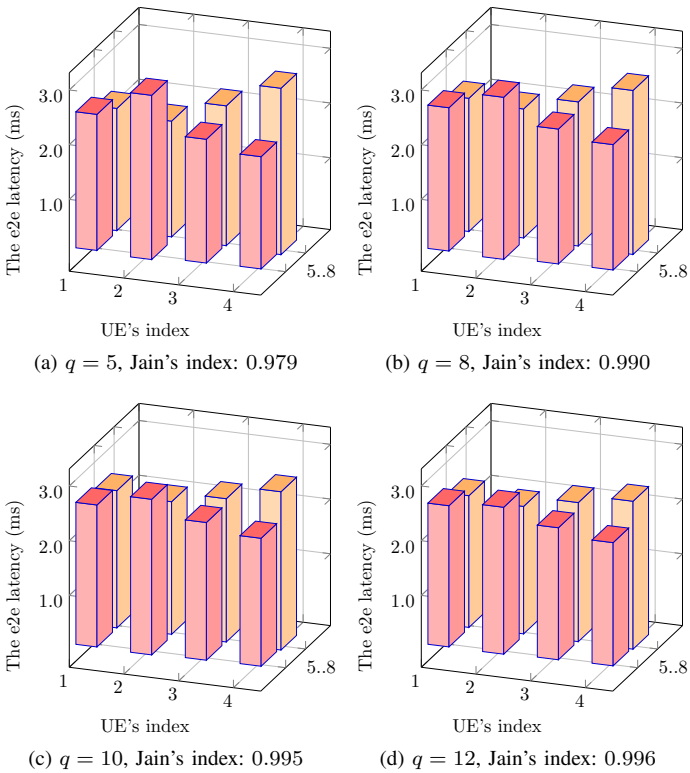


Fig. 3: Convergence behavior of Algorithms 1 and 2 with $M = 8$ UEs, $K = 2$ ESs and $q = \{8, 10\}$.



(a) $q = 5$, Jain's index: 0.979 (b) $q = 8$, Jain's index: 0.990

(c) $q = 10$, Jain's index: 0.995 (d) $q = 12$, Jain's index: 0.996

Fig. 4: The e2e latency distribution among 8 UEs: (a) $q = 5$, (b) $q = 8$, (c) $q = 10$, (d) $q = 12$ under Algorithm 2 with $\rho = 2$.

2) Impact of the fairness parameter q on the performance of Algorithm 2: We now investigate the impact of the fairness parameter in the distributed solution. To measure the fairness among UEs, we consider the well-known Jain's fairness index, which is given by

$$J(\tau_1, \tau_2, ..., \tau_M) = \frac{\left(\sum_{m=1}^{M} \tau_m\right)^2}{M \sum_{m=1}^{M} \tau_m^2} \tag{52}$$

This article has been accepted for publication in IEEE Journal on Selected Areas in Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2023.3310087

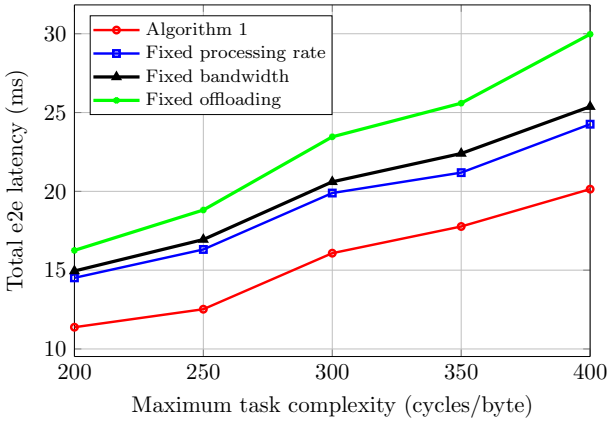


Fig. 5: Performance comparison between Algorithm 1 and baseline schemes versus the task complexity.

where τ_m is the optimum e2e latency of the m-th UE. Fig. 4 plots the e2e latency of all individual UEs for different values of the fairness parameter (q). As can be seen that when q varies from 5 to 12, the fairness index of UEs considerably increases which indicates the effectiveness of the fairness-aware design in the distributed solution. More specifically, when the fairness parameter is set to $q = 5$, the difference between the highest latency and the lowest among the latency distribution of 8 UEs in Fig. 4a is considerably large, which is reflected in the Jain's fairness index approximately 0.979. Meanwhile, Fig. 4d experiences a better fairness, where the Jain's index is improved to 0.996 with $q = 12$.

3) Performance comparison: In Fig 5, we illustrate the e2e latency vs the maximum task complexity $\eta_m \triangleq C_m/D_m, \forall m$ for different resource allocation schemes. As seen, the proposed solution with joint communication and computation variables outperforms baseline schemes in terms of the total e2e latency, especially when computational tasks become more complicated. For instance, when the maximum task complexity reaches to 400 cycles/byte, the proposed solution achieves lower latency than that in the fixed processing rate and the fixed offloading schemes about 5 ms and 10 ms, respectively. It is important to note that the fixed offloading scheme experiences the worst performance among the other baseline schemes, which clearly demonstrates that the optimisation of task offloading portions plays a vital roles in minimising the execution time of computational tasks in edge computing.

4) Consensus evolution of the distributed Algorithm 2: The impact of Lagrange parameter ρ on the consensus evolution of $\mathbf{f}^{\mathsf{ue}}_{mk}$ is investigated in Fig. 6. It can be observed that ρ has a strong impact on the total e2e latency as well as the consensus behaviour of the local processing rate of UEs. Algorithm 2 with $\rho = 2$ offers better total e2e latency, compared with other settings. In particular, the obtained total latency with $\rho = 2$ is smaller than the scheme with $\rho = 20$ by approximately 1 ms. In addition, Fig. 6 reveals that the consensus procedure works effectively for all values of ρ. In this regard, the difference between optimised values of UEs' local processing rate approaches zero at the convergence point, which confirms that the proposed distributed solution works properly in solving the original optimisation problem.

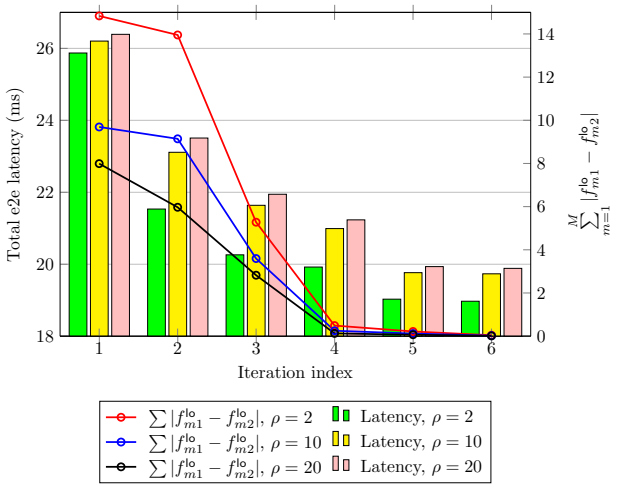


Fig. 6: The consensus evolution of $\mathbf{f}^{\mathsf{ue}}_{mk}$ over the iteration index with different values of ρ.

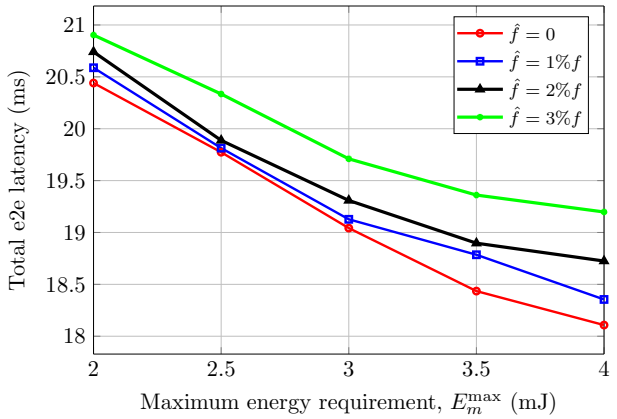


Fig. 7: The impact of UEs' energy budget and the deviation of processing rate.

5) Impacts of UEs' energy consumption budget: Fig. 7 illustrates the impacts of the UEs' energy consumption budget as well as the deviation value of processing rates. In particular, it can be clearly seen that when the maximum energy requirement of UEs ($E^{\max}_m$) increases, the total e2e latency of UEs significantly reduces. For instance, in the perfect processing rate estimation ($\hat{f} = 0$), the total e2e latency declines approximately 2 ms when $E^{\max}_m$ reaches 4 mJ. Additionally, Fig. 7 shows that the more accurately the DT estimates the processing rate, the better the performance can be obtained. In this regard, when the DT perfectly estimates the processing rate of UEs and ESs, the total e2e latency is smaller than the scheme with 3% deviation around 1 ms.

6) Impacts of task complexity: Fig. 8 investigates the impact of the task complexity on the system performance. We can see from the figure that the total latency steadily increases when the task complexity increases. For instance, in the scheme with $E^{\max}_m = 5$ mJ, the total latency experiences a steady rise from about 11 ms to 20 ms when the maximum task complexity reaches 400 cycles/byte. More interestingly, Fig. 8 also shows the offloading behaviour of UEs among different levels of task complexity. When the computational task becomes more and more complicated to

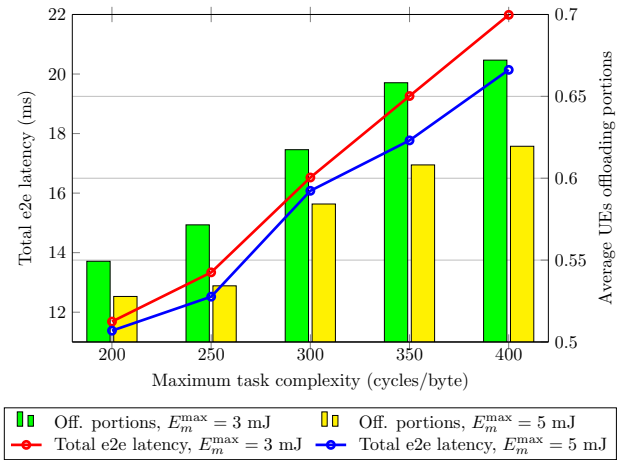


Fig. 8: The impact of the task complexity under different values of energy consumption budget.

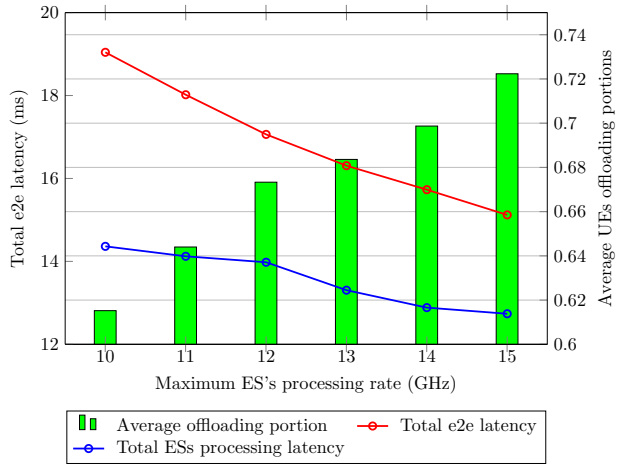


Fig. 9: The impact of ESs' processing rate and the offloading behaviour.

be processed, the offloading portion gradually increases to minimise the processing latency. In this regard, in the scheme with $E_m^{\max} = 3$ mJ, the average offloading portion rises by 12% when the maximum task complexity increases from 200 to 400 cycles/byte, which validates the effectiveness of the proposed solution for task offloading.

7) Impacts of ESs' processing rate: For the purpose of investigating the effect of ESs' computing capacity on latency reduction, we run simulations with different values of the maximum ESs' processing rate ($F_{\max}^{\text{es}}$). Fig. 9 illustrates the total e2e latency, the total ESs processing latency and the average offloading portion in the scenario of $M = 8$ UEs and $K = 2$ ESs. Firstly, it can be seen that when ESs are equipped with more computing power, the total e2e latency gradually declines. Specifically, the total e2e latency is decreased with 2 ms when the computing capacity of ESs increases from 10 to 15 GHz. Secondly, Fig. 9 also shows a similar trend in the total ESs' processing latency among different levels of the maximum ES's processing rate. Finally, the ES computing capacity increases from 10 to 15 GHz, the average offloading portion of UEs steadily rises from around 62% to over 72%. And again, these results demonstrate that the proposed solution for task offloading executes accurately to reduce latency.

VII. Conclusion and Future Work

In this paper, we have studied the distributed resource management in DT-assisted edge computing with with ultra-reliable and low latency communications. The addressed problem comprehensively optimises communication and computation resources, such as bandwidth allocation, transmission power, offloading policies, and processing rates of UEs and ESs. We have introduced a newly fairness-aware latency minimisation framework, which helps provide a fairer solution without the need of complicated design. To solve the formulated problem, both centralised and distributed approaches have been proposed to achieve at least a locally optimal solution. More importantly, the distributed solution with the ability of parallel processing not only reduces execution time but also has strong applicability to large-scale networks. Extensive simulation results have been provided to validate the effectiveness of the proposed solutions in terms of the convergence and the total e2e latency as well as demonstrating the impact of many involved parameters in the system.

For future works, a promising direction would be a deeper interaction of the DT concept with other edge computing problems, such as joint computation, communication and caching, popularity-aware task offloading and freshness-aware model, etc. In the development of more adaptive and intelligent solutions, machine learning-based approaches can be exploited to provide real-time solutions for resource management in the practical deployment of edge computing systems.

Appendix

Following inequality [9, eq. (73)], [46], we can show that

$$z\ln(1+\frac{x}{y}) \geq 2\bar{z}\ln(1+\frac{\bar{x}}{\bar{y}}) + \frac{\bar{z}\bar{x}}{\bar{x}+\bar{y}}\left(2-\frac{\bar{x}}{x}-\frac{y}{\bar{y}}\right) - \frac{\ln(1+\bar{x}/\bar{y})}{z}\bar{z}^2 \tag{53}$$

with $x>0, y>0, z>0$, and $(\bar{x},\bar{y},\bar{z})$ are the feasible point of (x,y,z). By letting $z = b_{mk}$, $\bar{z} = b_{mk}^{(i)}$, $x = p_{mk}(L-1)\sigma_{mk}^2$, $\bar{x} = p_{mk}^{(i)}(L-1)\sigma_{mk}^2$, $y = p_{mk}\delta_{mk}^2+1$, and $\bar{y} = p_{mk}^{(i)}\delta_{mk}^2+1$, the global lower bound of the function $G_{mk}(b_{mk},p_{mk})$ is given as

$$\begin{aligned} G_{mk}(b_{mk},p_{mk}) &\geq 2b_{mk}^{(i)}\ln\left(1+\frac{p_{mk}^{(i)}(L-1)\sigma_{mk}^2}{p_{mk}^{(i)}\delta_{mk}^2+1}\right) \\ &+ \frac{b_{mk}^{(i)}\left(p_{mk}^{(i)}(L-1)\sigma_{mk}^2\right)\left(2-p_{mk}^{(i)}/p_{mk}-f_{mk}^{(i)}\right)}{p_{mk}^{(i)}(L-1)\sigma_{mk}^2+(p_{mk}^{(i)}\delta_{mk}^2+1)} \\ &- \frac{\ln\left(1+p_{mk}^{(i)}(L-1)\sigma_{mk}^2/\left(p_{mk}^{(i)}\delta_{mk}^2+1\right)\right)(b_{mk}^{(i)})^2}{b_{mk}} \\ &\triangleq \mathcal{G}_{mk}^{(i)}(b_{mk},p_{mk}) \end{aligned} \tag{54}$$

where $f_{mk}^{(i)} = \frac{p_{mk}\delta_{mk}^2+1}{p_{mk}^{(i)}\delta_{mk}^2+1}$. It is noted that $\mathcal{G}_{mk}^{(i)}(b_{mk},p_{mk})$ is a concave function, satisfying $\mathcal{G}_{mk}^{(i)}\left(b_{mk}^{(i)},p_{mk}^{(i)}\right) = G_{mk}\left(b_{mk}^{(i)},p_{mk}^{(i)}\right)$.

Next, we apply the following inequality [9, eq. (75)]

$$\sqrt{x} \leq \frac{\sqrt{\bar{x}}}{2} + \frac{x}{2\sqrt{\bar{x}}} \tag{55}$$

with $x = b_{mk}$, $\bar{x} = b_{mk}^{(i)}$ to approximate $W_{mk}\left(b_{mk}\right)$ as

$$W_{mk}\left(b_{mk}\right) \leq \frac{Q^{-1}(\epsilon_{mk})}{\sqrt{\phi B}} \left(\frac{\sqrt{b_{mk}^{(i)}}}{2} + \frac{b_{mk}}{2\sqrt{b_{mk}^{(i)}}} \right) \triangleq \mathcal{W}_{mk}^{(i)}\left(b_{mk}\right) \tag{56}$$

which is a linear function in b_{mk}.

REFERENCES

[1] D. V. Huynh, V.-D. Nguyen, S. R. Khosravirad, and T. Q. Duong, "Fairness-aware latency minimisation in digital twin-aided edge computing with ultra-reliable and low-latency communications: A distributed optimisation approach (invited paper)," in *Proc. of 56th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 31–Nov. 02, 2022.

[2] K.-C. Chen, S.-C. Lin, J.-H. Hsiao, C.-H. Liu, A. F. Molisch, and G. P. Fettweis, "Wireless networked multirobot systems in smart factories," *Proc. IEEE*, vol. 109, no. 4, pp. 468–494, Apr. 2021.

[3] M. S. Elbamby, C. Perfecto, C.-F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proc. IEEE*, vol. 107, no. 8, pp. 1717–1737, Aug. 2019.

[4] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjoland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, Jul. 2021.

[5] 3GPP, "Study on scenarios and requirements for next generation access technologies," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.913, 2018, version 15.0.0.

[6] 3GPP, "Release 16 description," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 21.916, 2020, version 1.0.0.

[7] H. Ren, C. Pan, Y. Deng, M. Elkashlan, and A. Nallanathan, "Joint pilot and payload power allocation for massive-MIMO-enabled URLLC IIoT networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 816–830, May 2020.

[8] C. Sun, C. She, C. Yang, T. Q. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 402–415, Jan. 2019.

[9] A. A. Nasir, H. D. Tuan, H. Nguyen, M. Debbah, and H. V. Poor, "Resource allocation and beamforming design in the short blocklength regime for URLLC," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1321–1335, Feb. 2021.

[10] D. V. Huynh, S. R. Khosravirad, L. D. Nguyen, and T. Q. Duong, "Multiple relay robots-assisted URLLC for industrial automation with deep neural networks," in *Proc. of IEEE Global Telecommun. Conf.*, Madrid, Spain, Dec.7–11 2021.

[11] A. A. Nasir, "Min-max decoding-error probability-based resource allocation for a urllc system," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2864–2867, 2020.

[12] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[13] Z. Zhou, Z. Wang, H. Yu, H. Liao, S. Mumtaz, L. Oliveira, and V. Frascolla, "Learning-based URLLC-aware task offloading for Internet of health things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 396–410, Feb. 2021.

[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[15] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[16] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2131–2165, Fourthquarter 2021.

[17] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated learning: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 14–41, May 2022.

[18] D. Callegaro and M. Levorato, "Optimal edge computing for infrastructure-assisted UAV systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1782–1792, Feb. 2021.

[19] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 302–317, Jan. 2021.

[20] T. T. Vu, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz, and T. V. Nguyen, "Optimal energy efficiency with delay constraints for multi-layer cooperative fog computing networks," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3911–3929, Jun. 2021.

[21] J. Wang, D. Feng, S. Zhang, A. Liu, and X.-G. Xia, "Joint computation offloading and resource allocation for MEC-enabled IoT systems with imperfect CSI," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3462–3475, Mar. 2021.

[22] D.-B. Ha1, V.-T. Truong, and Y. Lee, "Intelligent reflecting surface assisted RF energy harvesting mobile edge computing NOMA networks: Performance analysis and optimization," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 9, no. 32, Aug. 2022.

[23] R. Lin, Z. Zhou, S. Luo, Y. Xiao, X. Wang, S. Wang, and M. Zukerman, "Distributed optimization for computation offloading in edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8179–8194, Dec. 2020.

[24] Y. Xiao and M. Krunz, "Distributed optimization for energy-efficient fog computing in the tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2390–2400, Nov. 2018.

[25] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: a survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13 789–13 804, Sep. 2021.

[26] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, Oct. 2020.

[27] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital twin assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, no. 2, pp. 1427–1444, Jan. 2022.

[28] T. Do-Duy, D. V. Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Commun. Lett*, vol. 11, no. 4, pp. 806–810, Apr. 2022.

[29] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[30] C. She, C. Yang, and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 72–78, Jun. 2017.

[31] M. Merluzzi, P. D. Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 342–356, Mar. 2020.

[32] D. V. Huynh, V.-D. Nguyen, S. R. Khosravirad, and T. Q. Duong, "Minimising latency for edge-cloud systems with ultra-reliable and low-latency communications," in *Proc. IEEE Int. Conf. Commun.*, Seoul, Korea, May16–20 2022.

[33] D. V. Huynh, V.-D. Nguyen, V. Sharma, O. A. Dobre, and T. Q. Duong, "Digital twin empowered ultra-reliable and low-latency communications-based edge networks in industrial IoT environment," in *Proc. IEEE Int. Conf. Commun.*, Seoul, Korea, May 16–20 2022.

[34] Y. Li, D. V. Huynh, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, "Unmanned aerial vehicles-aided edge networks with ultra-reliable low-latency communications: A digital twin approach," *IET Signal Processing*, vol. 16, no. 8, pp. 897–908, Oct. 2022.

[35] Q. Liu, T. Han, and N. Ansari, "Joint radio and computation resource management for low latency mobile edge computing," in *IEEE Global Telecommun. Conf.*, Abu Dhabi, United Arab Emirates, 2018.

[36] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[37] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[38] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, May 2010.

[39] V.-D. Nguyen, H. V. Nguyen, O. A. Dobre, and O.-S. Shin, "A new design paradigm for secure full-duplex multiuser systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1480–1498, Jul. 2018.

[40] V.-D. Nguyen, T. Q. Duong, H. D. Tuan, O.-S. Shin, and H. V. Poor, "Spectral and energy efficiencies in full-duplex wireless information and power transfer," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2220–2233, May 2017.

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, Jul. 2011.
[42] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia: MPS-SIAM Series on Optimi., SIAM, 2001.
[43] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Proc. of AFSS international conference on fuzzy systems*. Springer, 2002, pp. 288–300.
[44] K. C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3-A Matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Softw.*, vol. 11, pp. 545–581, Jan. 1999.
[45] C.-F. Liu, M. Bennis, and M. D. andH. Vincent Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.
[46] Z. Sheng, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Power allocation for energy efficiency and secrecy of wireless interference networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3737–3751, Jun. 2018.

Dang Van Huynh (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University Belfast, U.K.

Van-Dinh Nguyen (S'14-M'19) received the M.E. and Ph.D. degrees in electronic engineering from Soongsil University, Seoul, South Korea, in 2015 and 2018, respectively. Since Sept. 2022, he is an Assistant Professor at VinUniversity, Vietnam.

Saeed R. Khosravirad (Member, IEEE) is a Member of Technical Staff at Nokia Bell Labs. He received his Ph.D. degree in telecommunications in 2015 from McGill University, Canada.

George K. Karagiannidis (Fellow, IEEE) is currently a Professor with the School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece, and the Head of the Wireless Communications and Information Processing (WCIP) Group. He is also Faculty Fellow with the Cyber Security Systems and Applied AI Research Center, Lebanese American University.

Trung Q. Duong (Fellow, IEEE) is a Chair Professor in Telecommunications at Queen's University Belfast, UK. He also holds a prestigious Research Chair of Royal Academy of Engineering.