Digital Twin Empowered Ultra-Reliable and Low-Latency Communications-based Edge Networks in Industrial IoT Environment

Dang Van Huynh*, Van-Dinh Nguyen†, Vishal Sharma*, Octavia A. Dobre‡, and Trung Q. Duong*

*Queen's University Belfast, UK (e-mail: {dhuynh01, v.sharma, trung.q.duong}@qub.ac.uk)
†University of Luxembourg, Luxembourg (e-mail: dinh.nguyen@uni.lu)
‡Memorial University of Newfoundland, Canada (e-mail: odobre@mun.ca)

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838860

***Abstract*—We address the problem of minimising latency with computation offloading in digital twin wireless edge networks in industrial Internet-of-Things environment via ultra-reliable and low latency communications links. The minimised latency is obtained by jointly optimising both communication and computation variables, namely transmit power, user association of IoT devices, offloading portions, the processing rate of users and edge servers. To deal with this challenging problem, we propose an iterative algorithm based on alternating optimisation approach combined with inner convex approximation framework. Simulation results demonstrate the proposed algorithm's effectiveness in reducing the latency compared with other benchmark schemes.**

I. Introduction

Recent advances in communication technologies and powerful computation platforms open opportunities to implement a wide range of breakthrough applications, especially for time-sensitive services. In terms of communication perspective, 5G New Radio with ultra-reliable and low latency communications (URLLC) plays a vital role in the development and deployment of mission-critical applications, which require extremely high demands on reliability and low latency communications. In terms of computation views, mobile edge computing (MEC) technology leverages the powerful computation capacity of nearby edge servers to reduce the overall latency of services [1], [2]. Task offloading is a critical technique in the MEC architecture, which allows constrained devices to partially offload computational tasks to edge servers equipped with stronger processors to minimise the overall latency [3].

Recently, digital twin (DT) has emerged as promising technology creating the virtual twins of physical objects, which benefits many domains such as real-time management and industrial automation [4], [5]. Combined with MEC technology, DT opens opportunities and challenges attracting many active research groups to investigate this topic [6]–[9]. In particular, the offloading latency minimisation problem for DT edge networks was addressed in [6]. This problem was solved by applying the deep reinforcement learning (DRL) method to find optimal offloading decisions, while other wireless communication factors were not fully considered in this work. In [7], adaptive edge association for wireless DT networks was introduced. A DRL-based algorithm was also exploited to solve the DT migration problem in this paper. Again, wireless aspects such as transmit power, channel conditions as well as computation resources of users and edge servers were not entirely taken into account. The combined problem of MEC, URLLC and DT was firstly investigated in [8]. This work aimed to minimise the energy consumption by optimising user association, resource allocation and offloading portions under URLLC-based transmission. A deep learning (DL) architecture was utilised with the support of DT to find optimal resource allocation and offloading decisions. However, other computation parameters such as the processing rate of users and edge servers were not considered in this work. Thus far, designing effective solutions for task offloading in MEC with DT technology is still an open research issue, especially for mission-critical applications in industrial scenarios.

Against mentioned literature, this work proposes a joint communication and computation offloading in URLLC-based edge networks with DT that takes into account all the above issues. We aim to minimise the worst-case latency of task offloading by optimising user association, transmit power, the processing rate of IoT devices (IoTs), edge servers (ESs) and offloading portion. The problem is formulated based on the edge network architecture under the DT paradigm, while the wireless communications between IoTs and ESs are established via URLLC links, which is applicable in the context of industrial automation.

II. System Model and Problem Formulation

Fig. 1 presents a DT-empowering URLLC-based edge network architecture for industrial automation. The physical layer consists of many industrial IoTs and ESs. These physical devices connect via URLLC links to ensure stringent requirements on reliability and low-latency communications in factory automation scenarios.

1) URLLC-based MEC architecture: Let $\mathcal{M} = \{1, 2..M\}$ be the set of M IoTs and $\mathcal{K} = \{1, 2, ..K\}$ be the set of K ESs. Each ES is associated with an access

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.

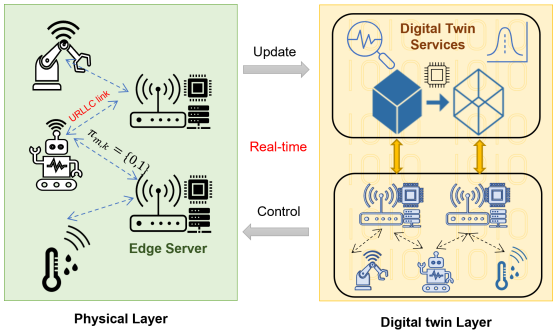

Fig. 1: Digital twin empowering URLLC-based edge network.

point (AP). The user association indicator is denoted as binary variables $\boldsymbol{\pi} = \{\pi_{mk}\}_{\forall m,k} = \{0,1\}_{\forall m,k}$; when $\pi_{mk} = 1$, there is a connection between m-th IoT and the k-th ES; otherwise, $\pi_{mk} = 0$. To guarantee performance, each ES only serves a maximum of $M_{\max}$ IoTs, and we have $\sum_{m\in\mathcal{M}} \pi_{mk} \leq M_{\max}, \forall k \in \mathcal{K}$.

2) Offloading in edge networks: A particular task from the m-th IoT device is represented by a tuple $J_m = \{D_m, C_m, T_m\}$, where D_m is data size (bits), C_m is required computation resource (cycles), and T_m (s) is the minimum required latency for task J_m.

Let $\boldsymbol{\alpha} = \{\alpha_m\}_{\forall m}$ be the amount of the task processed locally, and $\boldsymbol{\beta} = \{\beta_{mk}\}_{\forall m,k}$ be the offloading factors of the m-th IoT to the k-th ES, which satisfies $0 \leq \alpha_m \leq 1$, $0 \leq \beta_{mk} \leq 1$. By this way, with the J_m task offloaded by the m-th IoT device, we have $\alpha_m + \sum_{k\in K} \pi_{mk}\beta_{mk} = 1, \ \forall m$.

3) Digital twin model: The DT services fully replicate the physical devices, including the information of hardware configuration, historical data, and real-time operating states. The DT can interact with the physical system via real-time update and control mechanism. The DT of URLLC-based MEC model can be represented as

$$\mathrm{DT} = \{(\mathcal{M}, \tilde{\mathcal{M}}), (\mathcal{K}, \tilde{\mathcal{K}})\}, \tag{1}$$

where $\{\tilde{\mathcal{M}}, \tilde{\mathcal{K}}\}$ are the replica of physical network, including all IoTs and ESs. Based on real-time updated information from physical objects, digital services in the DT layer provides comprehensive functions to manage the system automatically.

In terms of devices DT, for the m-th IoT device, its DT (DT_m) can be expressed as

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

where f_m^{Io} is the estimated processing rate of the physical IoT device, and $\hat{f}_m^{\mathsf{Io}}$ is the deviation between the real device and its DT.

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

Similarly, for the k-th ES, its digital twin DT_k can be expressed as

$$\mathrm{DT}_k = (f_k^{\mathsf{es}}, \hat{f}_k^{\mathsf{es}}), \tag{3}$$

where f_k^{es} is the estimated processing rate of the physical server, and $\hat{f}_k^{\mathsf{es}}$ is the deviation between the real ES and DT.

The DT of ESs provides an estimated processing rate of ESs to reflect current states of the real ESs in terms of computation ability. This mechanism allows the DT to make a decision on adjusting offloading factors and edge selection policies to maximise the system performance.

A. Communication Model

1) Transmission model: Each AP is equipped with L antennas to serve M single-antenna IoTs. Let $\mathbf{h}_{km} \in \mathbb{C}^{L\times 1}$ be the channel vector between the k-th AP and the m-th IoT, can be modelled as $\mathbf{h}_{mk} = \sqrt{g_{mk}}\bar{\mathbf{h}}_{mk}$. Here, g_{mk} denotes the large-scale channel coefficient, including the pathloss and shadowing, and $\bar{\mathbf{h}}_{mk}$ is the small-scale fading following the distribution of $\mathcal{CN}(0, \mathbf{I})$. Let $\mathbf{H}_k \in \mathbb{C}^{L\times M}$ be the channel matrix from M devices to the k-th AP, with $\mathbf{H}_k = [\mathbf{h}_{k1}, \mathbf{h}_{k2}, .., \mathbf{h}_{kM}]$. Under the shared wireless medium, the $L \times 1$ received signal vector at the k-th AP is given by $\mathbf{y}_k = \sum_{m=1}^{M} \mathbf{h}_{km}\sqrt{p_m}s_m + \mathbf{n}_k$, where p_m is the payload power of the m-th device, s_m is the zero mean and unit variance Gaussian information message from the m-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.

To guarantee fairness among all IoTs, and further improve wireless transmission performance, we adopt the matched filtering and successive interference cancellation (MF-SIC) at the APs. By using MF-SIC, we assume that the decoding order follows IoTs' index by arranging the channel vector as $\|\mathbf{h}_{1k}\|^2 \geq \|\mathbf{h}_{2k}\|^2 \geq, \cdots, \geq \|\mathbf{h}_{Mk}\|^2, \forall k$. Then the signal-to-interference-plus-noise (SINR) at the k-th AP of the signal from the m-th IoT can be expressed as

$$\gamma_{mk}(\mathbf{p}, \boldsymbol{\pi}) = \frac{\pi_{mk}p_m||\mathbf{h}_{km}||^2}{\mathcal{I}_{mk}(\mathbf{p}, \boldsymbol{\pi}) + N_0}, \tag{4}$$

where $\mathcal{I}_{mk}(\mathbf{p}, \boldsymbol{\pi}) = \sum_{n>m}^{M} \pi_{nk}p_n||\mathbf{h}_{kn}||^2$ is the interference power caused by IoTs $n > m$.

2) URLLC uplink transmission rate: The approximation of achievable transmission rate in URLLC finite blocklength is [10], [11]:

$$R_{mk}^{ul}(\mathbf{p}, \boldsymbol{\pi}) \approx (1-\omega_k)\, B \log_2\left[1 + \gamma_{mk}(\mathbf{p}, \boldsymbol{\pi})\right] - B\sqrt{\frac{(1-\omega_k)\, V_{mk}(\mathbf{p}, \boldsymbol{\pi})}{N}}\frac{Q^{-1}(\epsilon)}{\ln 2}, \tag{5}$$

where $\omega_k = \sum_{m\in\mathcal{M}} \pi_{mk}/N, \forall k$, 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, $\gamma_{mk}(\mathbf{p}, \boldsymbol{\pi})$ 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 $V_{mk}(\mathbf{p}, \boldsymbol{\pi}) = 1 - [1 + \gamma_{mk}(\mathbf{p}, \boldsymbol{\pi})]^{-2}$. When the blocklength N approaches to infinity, the data rate R_{mk} approaches $(1-\omega_k)\, B \log_2(1 + \gamma_{mk}(\mathbf{p}, \boldsymbol{\pi}))$, which is the classic Shannon's equation.

Then the wireless transmission latency between the m-th IoT and the k-th ES for task offloading can be expressed as

$$T_{mk}^{cm}(\mathbf{p}, \boldsymbol{\pi}, \beta_{mk}) = \max_{\forall k\in\mathcal{K}} \left\{\frac{\beta_{mk}D_m}{R_{mk}^{ul}(\mathbf{p}, \boldsymbol{\pi})}\right\}. \tag{6}$$

B. Computation Model

1) Local processing: The m-th IoT executes α_m portion of task J_m with the estimated processing rate f_m, and the estimated time required to execute the task locally is given by

$$\tilde{T}_m^{lc}\left(\alpha_m, f_m^{\mathsf{lo}}\right) = \frac{\alpha_m C_m}{f_m^{\mathsf{lo}}}. \tag{7}$$

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:

$$\Delta T_m^{lc}\left(\alpha_m, f_m^{\mathsf{lo}}\right) = \frac{\alpha_m C_m \hat{f}_m^{\mathsf{lo}}}{f_m^{\mathsf{lo}}\left(f_m^{\mathsf{lo}} - \hat{f}_m^{\mathsf{lo}}\right)}. \tag{8}$$

The actual time for local computing is expressed as

$$T_m^{lc} = \Delta T_m^{lc} + \tilde{T}_m^{lc}. \tag{9}$$

2) Edge processing: Given the estimated processing rate of the k-th ES is f_k^{es}, the estimated latency of the k-th ES to execute task J_m is given by

$$\tilde{T}_{mk}^{ed}\left(\pi_{mk}, \beta_{mk}, f_k^{\mathsf{es}}\right) = \max_{\forall k \in \mathcal{K}} \left\{ \frac{\pi_{mk}\beta_{mk} C_m}{f_k^{\mathsf{es}}} \right\}. \tag{10}$$

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

$$\Delta T_{mk}^{ed}\left(\pi_{mk}, \beta_{mk}, f_k^{\mathsf{es}}\right) = \frac{\pi_{mk}\beta_{mk} C_m \hat{f}_k^{\mathsf{es}}}{f_k^{\mathsf{es}}\left(f_k^{\mathsf{es}} - \hat{f}_k^{\mathsf{es}}\right)}. \tag{11}$$

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

$$T_{mk}^{ed} = \Delta T_{mk}^{ed} + \tilde{T}_{mk}^{ed}. \tag{12}$$

C. Latency Model

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

$$T_m^{\mathsf{tot}} = T_m^{lc} + T_{mk}^{cm} + T_m^{ed} = \frac{\alpha_m C_m}{f_m^{\mathsf{lo}} - \hat{f}_m^{\mathsf{lo}}} + \max_{\forall k \in \mathcal{K}} \left\{ \frac{\pi_{mk}\beta_{mk} D_m}{R_{mk}^{ul}(\mathbf{p}, \boldsymbol{\pi})} \right\} + \max_{\forall k \in \mathcal{K}} \left\{ \frac{\pi_{mk}\beta_{mk} C_m}{f_k^{\mathsf{es}} - \hat{f}_k^{\mathsf{es}}} \right\}. \tag{13}$$

D. Energy Consumption Model

Total energy consumption of the m-th IoT includes energy for transmission and computation:

$$E_m^{\mathsf{tot}}\left(\alpha_m, \boldsymbol{\beta}, \mathbf{p}, \boldsymbol{\pi}\right) = E_m^{cp} + E_m^{cm} = \alpha_m \frac{\theta}{2} C_m \left(f_m^{\mathsf{lo}} - \hat{f}_m^{\mathsf{lo}}\right)^2 + \sum_{k=1}^{K} p_m \frac{\beta_{mk}\pi_{mk} D_m}{R_{mk}(\mathbf{p}, \boldsymbol{\pi})}, \tag{14}$$

where $\theta_m/2$ represents the average switched capacitance and the average activity factor of the m-th IoT [12].

E. Problem Formulation

Here, the worst-case of the total DT latency is minimised by optimising user association, offloading policies, transmit power, and estimated processing rates of IoTs and ESs. By defining the following notations $\mathscr{D} \triangleq \{\alpha_m, \beta_{mk}, \forall m, k | 0 \le \alpha_m \le 1, 0 \le \beta_{mk} \le 1, \forall m, k\}$, $\mathscr{P} \triangleq \{p_m, \forall m | 0 \le p_m \le P_m^{\max}, \forall m\}$, $\mathscr{F} \triangleq \{f_m^{\mathsf{lo}}, f_k^{\mathsf{es}}, \forall m, k | 0 \le f_m^{\mathsf{lo}} \le F_{\max}^{\mathsf{lo}}, \forall m; 0 \le f_k^{\mathsf{es}} \le F_{\max}^{\mathsf{es}}, \forall k\}$, and $\Pi \triangleq \{\pi_{mk}, \forall m, k | \pi_{mk} \in \{0, 1\}, \forall m, k\}$ as the set constraints of offloading decisions, uplink transmission power, processing rates and association policies, respectively, the problem is formulated as follows:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{p}, \mathbf{f}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{\mathsf{tot}}\left(\boldsymbol{\pi}, \alpha_m, \beta_{mk}, \mathbf{f}, \mathbf{p}\right) \right\}, \tag{15a}$$

$$\text{s.t. } T_m^{\mathsf{tot}}\left(\boldsymbol{\pi}, \alpha_m, \beta_{mk}, \mathbf{f}, \mathbf{p}\right) \le T_m^{\max}, \forall m, \tag{15b}$$

$$\sum_{m \in \mathcal{M}} \pi_{mk} \le M_{\max}, \forall m, \tag{15c}$$

$$\alpha_m + \sum_{k \in K} \pi_{mk}\beta_{mk} = 1, \tag{15d}$$

$$R_{mk}\left(\mathbf{p}, \boldsymbol{\pi}\right) \ge \pi_{mk} R_{\min}, \forall m, k, \tag{15e}$$

$$E_m^{\mathsf{tot}}\left(\boldsymbol{\pi}, \alpha_m, \boldsymbol{\beta}, \mathbf{p}\right) \le E_m^{\max}, \forall m, \tag{15f}$$

$$\sum_{m \in \mathcal{M}} \pi_{mk}\beta_{mk} f_k^{\mathsf{es}} \le F_{\max}^{\mathsf{es}}, \forall m, k, \tag{15g}$$

$$\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathscr{D}, \boldsymbol{\pi} \in \Pi, \mathbf{p} \in \mathscr{P}, \mathbf{f} \in \mathscr{F}, \tag{15h}$$

where constraint (15b) presents maximum latency constraint for every incoming task. Constraint (15c) means that each ES can serve a maximum of $M_{\max}$ IoTs. Constraints (15e) and (15f) are the minimum transmission rate requirement for uplink transmission and the maximum energy consumption requirement of IoTs, respectively. Finally, constraint (15g) ensures that the computation resources of ESs are not allocated in excess.

III. Proposed Solution

To solve problem (15), we replace the objective function by an upper bound function with introduced variables $\mathbf{t} \triangleq \{t_{lc}, t_{cm}, t_{ed}\}$ satisfying $\tau_m\left(t_{lc}, t_{cm}, t_{ed}\right) \triangleq t_{lc} + t_{cm} + t_{ed}$, and equivalently transforms (15) to

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{p}, \mathbf{f}, \mathbf{t}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \tag{16a}$$

$$\text{s.t.} \quad (15c) - (15h) \tag{16b}$$

$$\tau_m(\mathbf{t}) \le T_m^{\max}, \forall m, \tag{16c}$$

$$t_{lc} \ge \frac{\alpha_m C_m}{f_m^{\mathsf{lo}} + \hat{f}_m^{\mathsf{lo}}}, \forall m, \tag{16d}$$

$$t_{cm} \ge \frac{\pi_{mk}\beta_{mk} D_m}{R_{mk}^{ul}(\mathbf{p}, \boldsymbol{\pi})}, \forall m, k \tag{16e}$$

$$t_{ed} \ge \frac{\pi_{mk}\beta_{mk} C_m}{f_k^{\mathsf{es}} + \hat{f}_k^{\mathsf{es}}}, \forall m, k. \tag{16f}$$

Due to the complexity of the non-convex problem (16) and strong coupling among optimisation variables, we decompose (16) into three sub-problems and solve the problem by combination of alternating optimisation (AO) approach and inner approximation (IA) framework (AO-IA) [13], [14]. The following subsections fully present the development of our proposed solution.

A. Association Optimisation

In this sub-problem, we solve the problem (16) with fixed values of $(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \mathbf{f}^{(i)}, \mathbf{p}^{(i)})$ to find the next iterative solution of user association.

$$\min_{\boldsymbol{\pi}, \mathbf{t}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \tag{17a}$$

$$\text{s.t.} \quad (15c) - (15h), (16c), (16e), (16f), \tag{17b}$$

As we can observe from (17), constraints (15e), (15f), and (16e) are non-convex. We are now in the position to approximate these constraints.

Convexify of (15e): To address the non-convex constraint (15e), we first rewrite that $\gamma_{mk}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) = \frac{\pi_{mk}}{q_{mk}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right)}$, where $q_{mk}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right)$ is defined as

$$q_{mk}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) \triangleq \frac{\mathcal{I}_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}) + N_0}{p_{mk}^{(i)}||\mathbf{h}_{km}||^2}. \tag{18}$$

Following the Appendix, we have

$$\begin{aligned} &R_{mk}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) \geq R_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) \\ &\triangleq \frac{(1-\omega_k)\,B}{\ln 2}\left[\mathcal{G}_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) - \kappa \mathcal{V}_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right)\right] \end{aligned} \tag{19}$$

under the trusted regions

$$q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m) + \pi_{mk} \leq 2\big(q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}\big), \tag{20}$$

$$\frac{q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m) + \pi_{mk}}{q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}} \leq 2\frac{q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m)}{q_m(\mathbf{p}^{(i+1)}, \boldsymbol{\pi}_m^{(i)})}, \tag{21}$$

where $\mathcal{G}_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right)$ and $\mathcal{V}_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right)$ are defined as (36) and (37) in the Appendix, $\kappa = \frac{Q^{-1}(\epsilon)}{\sqrt{(1-\omega)N}}$. As a results, we innerly approximate constraint (15e) as

$$R_{mk}^{(i)}\left(\mathbf{p}^{(i)}, \boldsymbol{\pi}\right) \geq \pi_{mk} R_{\min}, \forall m, k. \tag{22}$$

Convexify of (15f): By introducing new variables $\hat{\mathbf{r}} \triangleq \{\hat{r}_{mk}\}_{\forall m,k}$ that satisfy $\frac{1}{R_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi})} \leq \hat{r}_{mk}$, (15f) is now equivalent to

$$\begin{cases} \alpha_m^{(i)} \dfrac{\theta}{2} C_m\big(f_m^{\mathsf{lo}}\big)^2 + D_m \displaystyle\sum_{k=1}^{K} \beta_{mk}^{(i)} \pi_{mk} \hat{r}_{mk} \leq E_m^{\max}, & \text{(23a)} \\ \dfrac{1}{R_{mk}^{ul(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi})} \leq \hat{r}_{mk}, \forall m, k. & \text{(23b)} \end{cases}$$

The constraint (23b) is now convex while (23a) is still non-convex. We follow this inequality

$$xy \leq \frac{1}{2}\left(\frac{\bar{y}}{\bar{x}}x^2 + \frac{\bar{x}}{\bar{y}}y^2\right), \tag{24}$$

with $x = \pi_{mk}$, $\bar{x} = \pi_{mk}^{(i)}$, $y = \hat{r}_{mk}$, $\bar{y} = \hat{r}_{mk}^{(i)}$ to approximate (23a) as

$$\begin{aligned} \sum_{k=1}^{K} D_m \beta_{mk}^{(i)} \frac{1}{2}\left(\frac{\hat{r}_{mk}^{(i)}}{\pi_{mk}^{(i)}}\pi_{mk}^2 + \frac{\pi_{mk}^{(i)}}{\hat{r}_{mk}^{(i)}}\hat{r}_{mk}^2\right) \\ + \alpha_m^{(i)} \frac{\theta}{2} C_m\big(f_m^{\mathsf{lo}}\big)^2 \leq E_m^{max}, \forall m, \end{aligned} \tag{25}$$

which is now a convex constraint.

Convexify of (16e): By using variables $\hat{\mathbf{r}}$ as defined in (23b), we rewrite (16e) as follow

$$t_{cm} \geq \beta_{mk}^{(i)} D_m \pi_{mk} \hat{r}_{mk}, \forall m \in \mathcal{M}, k \in \mathcal{K}. \tag{26}$$

We apply (24) with $x = \pi_{mk}$, $\bar{x} = \pi_{mk}^{(i)}$, $y = \hat{r}_{mk}$, $\bar{y} = \hat{r}_{mk}^{(i)}$, to approximate (26) as

$$t_{cm} \geq \beta_{mk}^{(i)} D_m \frac{1}{2}\left(\frac{\hat{r}_{mk}^{(i)}}{\pi_{mk}^{(i)}}\pi_{mk}^2 + \frac{\pi_{mk}^{(i)}}{\hat{r}_{mk}^{(i)}}\hat{r}_{mk}^2\right), \forall m, k, \tag{27}$$

which is now a convex constraint.

Based on the above developments, we solve the following approximate convex program of (17) at iteration i-th:

$$\min_{\boldsymbol{\pi}, \hat{\mathbf{r}}, \mathbf{t}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \tag{28a}$$

$$\text{s.t.} \quad (15c), (15d), (15g), (15h), (16c), (16f), (20), (21), (22), (23b), (25), (27). \tag{28b}$$

This is a convex problem now, and can be solved effectively with standard convex solvers such as CVX [15] or CVXPY [16].

B. Offloading Policies Optimisation

In this subsection, we solve (16) with fixed $(\mathbf{p}^{(i)}, \mathbf{f}^{(i)}, \boldsymbol{\pi}^{(i+1)})$ to find next optimal values $(\boldsymbol{\alpha}^{(i+1)}, \boldsymbol{\beta}^{(i+1)})$

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{t}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \tag{29a}$$

$$\text{s.t.} \quad (15d), (15f), (15g), (15h), (16c) - (16f). \tag{29b}$$

The sub-problem (29) is obviously a convex program with all linear constraints, which can be solved by CVX [15].

C. Computation and Communication Resource Optimisation

In this subproblem, we solve (16) with given $(\boldsymbol{\pi}^{(i+1)}, \boldsymbol{\alpha}^{(i+1)}, \boldsymbol{\beta}^{(i+1)})$ to find the next optimal values $(\mathbf{p}^{(i+1)}, \mathbf{f}^{(i+1)})$

$$\min_{\mathbf{p}, \mathbf{f}, \mathbf{t}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \tag{30a}$$

$$\text{s.t.} \quad (15e) - (15h), (16c) - (16f). \tag{30b}$$

As can be observed from the sub-problem (30), the constraints (15e), (15f), (16e) are non-convex. We will address these constraints to transform the sub-problem (30) into a convex program.

Convexify of (15e): To address this constraint, we process similarly as in subsection III-A with the inner approximated rate $R_{mk}^{(i)}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right)$ constructed from $q_{mk}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right) \triangleq \frac{\mathcal{I}_{mk}(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}) + N_0}{\pi_{mk}^{(i)}||\mathbf{h}_{km}||^2}$, $\mathcal{G}_{mk}^{(i)}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right)$, and $\mathcal{V}_{mk}^{(i)}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right)$. Consequently, the constraint (15e) can be iteratively replaced by

$$R_{mk}^{(i)}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right) \geq \pi_{mk} R_{\min}, \forall m, k. \tag{31}$$

Convexify of (15f): By introducing new variables $\check{\mathbf{r}} \triangleq \{\check{r}_m\}_{\forall m}$ that satisfy $\frac{1}{R_{mk}} \leq \check{r}_m$, then (15f) is equivalent to

$$\begin{cases} \alpha_m^{(i)} \dfrac{\theta}{2} C_m\big(f_m^{\mathsf{lo}}\big)^2 + \displaystyle\sum_{k=1}^{K} \beta_{mk}^{(i)} \pi_{mk}^{(i)} D_m p_m \check{r}_m \leq E_m^{\max}, & \text{(32a)} \\ \dfrac{1}{R_{mk}^{(i)}\left(\mathbf{p}, \boldsymbol{\pi}^{(i+1)}\right)} \leq \check{r}_m, \forall m, k. & \text{(32b)} \end{cases}$$

The constraint (32a) is still non-convex so we follow (24) with $x = p_m$, $y = \check{r}_m$, $\bar{x} = p_m^{(i)}$, $\bar{y} = \check{r}_m^{(i)}$ to iteratively express (32a) as

$$\begin{aligned} \sum_{k=1}^{K} \beta_{mk}^{(i)} \pi_{mk}^{(i)} D_m \frac{1}{2}\left(\frac{\check{r}_m^{(i)}}{p_m^{(i)}}p_m^2 + \frac{p_m^{(i)}}{\check{r}_m^{(i)}}\check{r}_m^2\right) \\ + \alpha_m^{(i)} \frac{\theta_m}{2} C_m\big(f_m^{\mathsf{lo}}\big)^2 \leq E_m^{max}, \forall m. \end{aligned} \tag{33}$$

which is now a convex constraint.

Convexify of (16e): By using $(\check{\mathbf{r}})$ defined in (32b), (16f) can be convexified as

$$t_{cm} \geq D_m \pi_{mk}^{(i+1)} \beta_{mk}^{(i+1)} \check{r}_{mk}. \quad (34)$$

Based on the above developments, we solve the following approximate convex program of (30) at iteration i-th:

$$\min_{\mathbf{p},\mathbf{f},\mathbf{t},\check{\mathbf{r}}} \quad \max_{\forall m \in \mathcal{M}} \{\tau_m(\mathbf{t})\}, \quad (35a)$$

$$\text{s.t.} \quad (15g), (15f), (16c) - (16f), (31), (32b), (33), (34). \quad (35b)$$

D. Proposed Algorithm

Let us define $\mathcal{S}_1^{(i)} \triangleq (\boldsymbol{\pi}^{(i)}, \hat{\mathbf{r}}^{(i)})$, $\mathcal{S}_2^{(i)} \triangleq (\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)})$, and $\mathcal{S}_1^{(i)} \triangleq (\mathbf{p}^{(i)}, \mathbf{f}^{(i)}, \check{\mathbf{r}}^{(i)})$. The overall algorithm for solving problem (15) is summarised in Algorithm 1.

Algorithm 1 : AO-IA based Algorithm for Solving (16).

1: **Input**: Set $i = 0$ and randomly choose initial feasible points $\mathcal{S}_1^{(0)}$, $\mathcal{S}_2^{(0)}$ and $\mathcal{S}_3^{(0)}$ to constraints in (28), (29), (35)
Set the tolerance $\varepsilon = 10^{-3}$ and the maximum number of iterations $I^{\max} = 20$.
2: **Repeat**
3: Solve problem (28) for given $\mathcal{S}_2^{(i)}, \mathcal{S}_3^{(i)}$ to obtain the optimal solution of $(\boldsymbol{\pi}^*, \hat{\mathbf{r}}^*)$ and update $\mathcal{S}_1^{(i+1)} := (\boldsymbol{\pi}^*, \hat{\mathbf{r}}^*)$;
4: Solve problem (29) with given $\mathcal{S}_1^{(i+1)}, \mathcal{S}_3^{(i)}$ to obtain the optimal solution of $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ and update $\mathcal{S}_2^{(i+1)} := (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$;
5: Solve problem (35) with given $\mathcal{S}_1^{(i+1)}, \mathcal{S}_2^{(i+1)}$ to obtain the optimal solution of $(\mathbf{p}^*, \mathbf{f}^*, \check{\mathbf{r}}^*)$ and update $(\mathcal{S}_3^{(i+1)} := (\mathbf{p}^*, \mathbf{f}^*, \check{\mathbf{r}}^*)$;
6: Set $i := i + 1$;
7: **Until** Convergence or $i > I^{\max}$.
8: Recover binary values of $\boldsymbol{\pi}^*$: $\pi_{mk}^* = \lfloor \pi_{mk}^{(i)} + 0.5 \rfloor_{\forall m,k}$;
9: Repeat from **Step 2** to **Step 7** with fixed $\boldsymbol{\pi}^*$;
10: **Output**: $\{\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\pi}^*, \mathbf{p}^*, \mathbf{f}^*\}$ and $\max\{\tau_m(\mathbf{t})\}_{\forall m}$.

IV. Numerical Results

A. Simulations Setting

We consider a small-scale scenario for factory automation where all APs (ESs) and IoTs are located within an area of 100m×100m [12]. The large-scale fading of the channel between the m-th IoT to the k-th AP is modelled as $g_{mk} = 10^{\mathbf{PL}(d_{mk})/10}$, where $\mathbf{PL}(d_{mk}) = -35.3 - 37.6 \log_{10} d_{mk}$ denotes the path loss in dB, which is a function of the distance d_{mk} [17]. The URLLC decoding error probability is set to $\epsilon = 10^{-9}$ [10]. Other parameters are summarised as follows [11], [12], [18]: number of antennas: $L = 8$, maximum transmit power: 23 dBm, bandwidth: $B = 10$ MHz, transmission duration URLLC: $\delta = 0.02$ ms, noise spectral density: -174 dBm/Hz, number of IoTs: $M = [8, 10]$, number of ESs: $K = 2$, maximum IoTs' processing rate: 3 GHz, total ES processing rate: $F_{\max}^{\mathsf{es}} = 8$ GHz, maximum IoTs each ES serves: $M_{\max} = 7$, IoTs input data size: $D_m^i = 100$ kB, required computation resource: $C_m = 960 \times 10^6$ cycles, total delay requirement: $T_m^{max} = 2$ s, minimum data rate: $R_{min}^{ul} = 1$ Mbps, maximum energy consumption: $E_m^{max} = 1$ Joule, effective capacitance coefficient $\theta_m = 10^{-27}$ Watt.s^3/cycle^3.

B. Numerical Results and Discussions

1) Impact of required computation resource: To demonstrate the effectiveness of the proposed solution in reducing latency, we compare the worst-case latency of the proposed algorithm with other benchmark schemes. In particular, Fig. 2 plots the worst-case latency among different values of required computation resource (C_m) under the same scenarios of $M = 10$ IoTs and $K = 2$ ESs. Unsurprisingly, the latency of all considered schemes raises when C_m increases. Importantly, the proposed algorithm always has a better performance compared with other schemes.

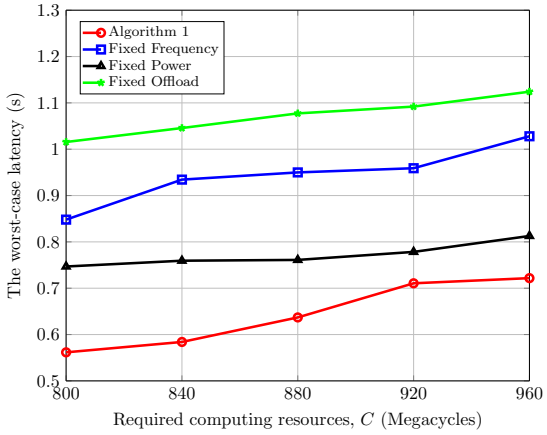

Fig. 2: The worst-case latency for different values of required computation resource ($C_m \triangleq C$) in the scenario of $M = 10, K = 2$, with $E_{\max} = 1$ Joule, $\hat{f}_m^{\mathsf{lo}} = \hat{f}_k^{\mathsf{es}} = 0$.

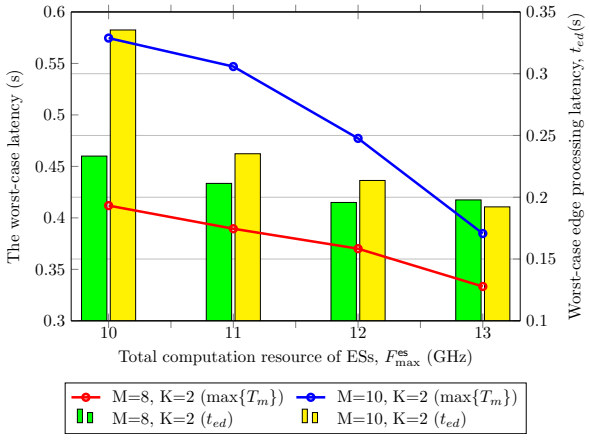

Fig. 3: The worst-case latency for the range of total computation resource of ESs in the scenarios of $M = 8, K = 2$ and $M = 10, K = 2$, with $E_{\max} = 1$ Joule.

2) Impact of ES processing rate: In the MEC architecture, the computation resource of ESs has a strong impact on the system performance. To verify this, Fig. 3 shows the worst-case latency among different values of the maximum processing rate of ESs. The figure clearly indicates that when the computation resource of ESs increases, both overall worst-case latency and edge server processing latency gradually

reduce. For example, with the scenarios of $M = 10, K = 2$, the worst-case latency reduces by approximately 200 ms when the total computation resource of ESs climbs to 26 GHz. These results validate that our intelligent tasks offloading solution works effectively.

V. Conclusion

We investigated the digital twin-aiding computation offloading in URLLC-based wireless edge networks. The addressed problem took into account various factors in both communication and computation variables in the system. To solve the problem, we proposed the AO-IA based algorithm dealing with three decomposed subproblems, namely user association, offloading policies optimisation, and resource optimisation. The effectiveness of the proposed solution was demonstrated through intensive numerical results.

Appendix

We first rewrite the SINR of IoT m as $\gamma_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) = p_m^{(i)}/q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)$. By applying the inequality [17, Eq. (72)] for $x = p_m$, $y = q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)$, $\bar{x} = p_m^{(i)}$, and $\bar{y} = q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})$, we have

$$G_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) \geq a_{mk}^{(i)} - \frac{b_{mk}^{(i)}}{\pi_m} - c_{mk}^{(i)} q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) \triangleq \mathcal{G}_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) \quad (36)$$

where $a_{mk}^{(i)} = \ln\big(1 + \frac{\pi_{mk}^{(i)}}{q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}\big) + 2\frac{\pi_{mk}^{(i)}}{\pi_{mk}^{(i)} + q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}$, $b_m^{(i)} = \frac{(\pi_{mk}^{(i)})^2}{p_m^{(i)} + q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_{mk})}$ and $c_{mk}^{(i)} = \frac{\pi_m^{(i)}}{\big(q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)}\big) q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}$.

To find an upper bounding convex function approximation of $V_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)$, we apply the inequality [17, Eq. (75)] for $x = 1 - 1/\big(1 + \gamma_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)\big)^2$ and $\bar{x} = 1 - 1/\big(1 + \gamma_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})\big)^2$, yielding

$$V_{mk} \leq \mathcal{V}_{mk}^{(i)} \triangleq d_{mk}^{(i)} - \frac{e_{mk}^{(i)} q_{mk}^2(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}{\big(q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)}\big)^2} \quad (37)$$

where

$$d_{mk}^{(i)} = 0.5\sqrt{V_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})} + 0.5/\sqrt{V_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})} \quad (38)$$

$$e_m^{(i)} = 0.5/\sqrt{V_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})}. \quad (39)$$

The function $\frac{q_{mk}^2(\mathbf{p}, \boldsymbol{\pi}_m^{(i)})}{\big(q_m(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)}\big)^2}$ in (37) is still not convex [17], and can be further approximated by using the inequalities [17, Eq. (77)] and [17, Eq. (76)] as

$$\frac{q_{mk}^2(\mathbf{p}, \boldsymbol{\pi}_m^{(i)})}{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)}} \frac{1}{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)}} \geq \frac{2}{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}} \left(\frac{2q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)}) q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}} - \frac{q_{mk}^2(p_m^{(i)})}{(q_{mk}(p_m^{(i)}) + \pi_{mk}^{(i)})^2} \right.$$

$$\left. (q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + p_m^{(i)}) \right) - \frac{q_{mk}^2(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m)}{(q_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}^{(i)})^2}$$

over the trusted regions defined in (20) and (21). By substituting this result to (37), yields

$$\mathcal{V}_{mk}^{(i)}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) \triangleq d_{mk}^{(i)} - \frac{2e_{mk}^{(i)}}{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}} \times \Big(2f_{mk}^{(i)} q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) - (f_{mk}^{(i)})^2 \big(q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m) + \pi_{mk}\big)\Big) + \frac{(f_{mk}^{(i)})^2}{q_{mk}^2(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})} q_{mk}^2 \text{ where } f_{mk}^{(i)} \triangleq \frac{q_{mk}(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)})}{q_m(\mathbf{p}^{(i)}, \boldsymbol{\pi}_m^{(i)}) + \pi_{mk}^{(i)}}.$$

References

[1] J. Zhang *et al.*, "Industrial pervasive edge computing-based intelligence iot for surveillance saliency detection," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5012–5020, Jul. 2021.

[2] C. Wang *et al.*, "Industrial cyber-physical systems-based cloud IoT edge for federated heterogeneous distillation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5511–5521, Aug. 2021.

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

[4] 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.

[5] 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.*, 2022, accepted.

[6] 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.

[7] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–1, 2021.

[8] 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.

[9] 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.*, vol. 4662, no. c, pp. 1–1, 2021.

[10] 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.

[11] 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.

[12] C.-F. Liu, M. Bennis, M. Debbah, and H. V. 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.

[13] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, July-Aug. 1978.

[14] 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.

[15] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[16] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[17] 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.

[18] 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.