Digital twin-aided intelligent offloading with edge selection in mobile edge computing

Do-Duy, T., Huynh, D. V., Dobre, O. A., Canberk, B., & Duong, T. Q. (2022). Digital twin-aided intelligent offloading with edge selection in mobile edge computing. *IEEE Wireless Communications Letters*, *11*(4), 806-810. https://doi.org/10.1109/LWC.2022.3146207

Published in:
IEEE Wireless Communications Letters

Document Version:
Peer reviewed version

Digital Twin-aided Intelligent Offloading with Edge Selection in Mobile Edge Computing

Tan Do-Duy, Dang Van Huynh, Octavia A. Dobre, Berk Canberk, and Trung Q. Duong

***Abstract*—In this paper, we study a mobile edge computing (MEC) architecture with the assistance of digital twin (DT) applied for industrial automation where multiple Internet-of-Things (IoT) devices intelligently offload computing tasks to multiple MEC servers to reduce end-to-end latency. To do so, first we propose and formulate a practical end-to-end latency minimisation problem in the DT-assisted MEC model subject to the constraints of quality-of-services and computation resource at the IoT devices and MEC servers in industrial IoT networks. Then, we solve the proposed latency minimisation problem by iteratively optimising the transmit power of IoT devices, user association, intelligent task offloading, and estimated CPU processing rate of the devices. Finally, simulation results are conducted to prove the effectiveness of the proposed method in terms of the latency performance compared with some conventional methods.**

***Index Terms*—Mobile Edge Computing, Digital Twin, IoT.**

I. Introduction

Mobile edge computing (MEC) has been recently considered as one of the promising solutions for Internet-of-Things (IoT) devices (e.g., sensors, smartphones, etc.) to reduce end-to-end latency by offloading their computing tasks to surrounding macro base stations (MBS) equipped with powerful computing resources [1], [2]. However, when large-scale scenarios with the heterogeneous deployment of IoT devices (UEs) and edge servers in the MEC system are considered, the challenges in designing the optimal offloading strategy with efficient resource allocation grow significantly due to the network size and dynamics. As a potential digital mapping technology, digital twin (DT) brings an excellent solution for intelligent resource allocation and network management in the MEC system by creating a real-time digital representation of the physical equipment [1]. By combining MEC and DT, the network status information can be efficiently monitored in real time and then provided directly to the decision-making module in the network in a centralised viewpoint.

There are already existing works focusing on designing optimal offloading schemes in MEC system assisted by the DT. For instance, in [1], the authors proposed a mobile offloading scheme to minimize the average offloading latency under the constraint of long-term migration cost when a mobile user (MU) offloads its computing task to nearby edge servers. The DT estimates the states of the edge servers and training data for offloading decisions. In another work, the authors in [3] employed the DT system as an assistant to help MUs in selecting high-quality MEC servers where DT manages the real-time status of the network so MUs can offload computing tasks to the MEC servers with low power expenditure and latency. Most recently, in [4], the authors exploited DT to support aerial-assisted internet of vehicles networks to capture the dynamic characteristics of the resource demands. Then, two incentive mechanisms were designed for jointly optimising the satisfaction of vehicles and the overall energy efficiency of road side units in the system.

Different from the existing works, in this paper we consider a DT empowered MEC architecture for industrial automation with multiple MEC servers to offload computing tasks from UEs. In particular, we formulate a practical end-to-end latency minimisation problem in the MEC model with the support of the DT technology. Then, we solve the proposed problem by iteratively optimising the transmit power of UEs, user association, intelligent task offloading, and estimated CPU processing rate of the DT. By means of numerical results, we show the effectiveness of our proposed method for solving the resource allocation issue under quality-of-service (QoS) constraints for dealing with the challenge of limited resources in IoT systems.

II. System model and problem formulation

In this paper, we consider a DT-empowered MEC architecture, where the physical layer consists of many UEs and MEC servers. Each UE can transmit data to multiple MECs to offload a computing task, and each MEC server can assist a finite number of UEs to guarantee performance. The association between IoT and MEC is established based on an edge selection indicator. The DT layer provides services which replicate the physical objects, data analysis, estimation, and decision making to manage and control physical system.

1) MEC architecture: There are M UEs, $\mathcal{M} = \{1, 2..M\}$ and K MEC servers, $\mathcal{K} = \{1, 2, ..K\}$. Each MEC server is associated with an access point (AP). The user association indicator is represented by the binary variable $\boldsymbol{\pi} = \{\pi_{mk}\}_{\forall m,k} = \{0, 1\}$; when $\pi_{mk} = 1$, there is a connection between the m-th IoT and the k-th MEC server. Each MEC server only assists a maximum of M_{max} UEs; we have $\sum\limits_{m \in \mathcal{M}} \pi_{mk} \leq M_{max}$.

2) Offloading in MEC: A particular task from the m-th UE is represented by a tuple $I_m = \{D_m, C_m, T_m\}$, where

T. Do-Duy is with Ho Chi Minh City University of Technology and Education, Vietnam (e-mail: tandd@hcmute.edu.vn). D. V. Huynh and T Q. Duong are with Queens University Belfast, UK (e-mail: {dhuynh01, trung.q.duong}@qub.ac.uk). O. A. Dobre is with Memorial University, Canada (e-mail: odobre@mun.ca). B. Canberk is with Istanbul Technical University, Turkey (e-mail: canberk@itu.edu.tr). The work of Tan Do-Duy was supported by the Ho Chi Minh City University of Technology and Education, Vietnam, under Project T2021-44T. 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.

D_m is the data size (bits), C_m is the required computation resource (cycles), and T_m is the minimum required latency for task I_m (seconds). 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 factor of the m-th UE to the k-th MEC server, which satisfy $0 \leq \alpha_m \leq 1$, $0 \leq \beta_{mk} \leq 1$. Given that the I_m task originated from the m-th UE, we have $D_m = \alpha_m D_m + \sum\limits_{k \in K} \pi_{mk}\beta_{mk}D_m$ and $C_m = \alpha_m C_m + \sum\limits_{k \in K} \pi_{mk}\beta_{mk}C_m$, where $\alpha_m + \sum\limits_{k \in K} \pi_{mk}\beta_{mk} = 1$.

3) Digital twin model: The DT services fully replicate the physical UEs, and include the information on the hardware configuration, historical data, and real-time operating states. The DT can interact with the physical system via a real-time update and control mechanism, which is represented as $DT = \{(\mathcal{M}, \tilde{\mathcal{M}}), (\mathcal{K}, \tilde{\mathcal{K}})\}$, where $\{\tilde{\mathcal{M}}, \tilde{\mathcal{K}}\}$ are the replica of the physical network including all UEs and MEC servers. For the m-th UE, its DT counterpart DT_m can be expressed as $DT_m = (f_m, \tilde{f}_m)$, where f_m and $\tilde{f}_m$ are the estimated CPU frequency assigned to the local task in the physical UE and the deviation between the value in the real UEs and its DT, respectively [1]. In particular, UEs can locally execute its tasks with the processing rate f_m and partially offload the tasks to the MEC servers to minimise computing latency. The DT layer has the estimated processing rate $\tilde{f}_m$ to replicate the behaviours of UEs and trigger decisions on optimising the physical UEs configuration. Similarly, for the k-th MEC servers, its DT counterpart DT_k can be expressed as $DT_k = (f_k, \tilde{f}_k)$, where f_k and $\tilde{f}_k$ are the estimated CPU frequency assigned to the local task in the physical MEC server and the deviation between the value in the real MEC server and its DT, respectively. The DT of MEC servers provides the estimated processing rate to reflect the current states of the real MEC servers in terms of computation ability. This mechanism allows the DT to make decision on adjusting offloading factors and edge selection policies to maximise the system performance.

A. Communication model between physical objects

Connections between UEs and MEC servers are established based on wireless communications. In this paper, we employ the efficient maximal ratio transmission (MRT) in beamforming design for the massive MIMO AP, which is formulated as [5] $\mathbf{f}_{mk} = \frac{\mathbf{g}^*_{mk}}{\|\mathbf{g}_{mk}\|}$, where $\mathbf{g}_{mk}$ is the channel coefficients between the m-th UE and the k-th AP; $\mathbf{f}_{mk}$ is the beamforming vector at the k-th AP. Here, we consider that the link between the m-th UE and the k-th MEC server includes both large-scale and small-scale fading effects as $\mathbf{g}_{mk} = \sqrt{\gamma_{mk}}\mathbf{h}_{mk}$, where γ_{mk} and $\mathbf{h}_{mk}$ are the path-loss expression and the small-scale fading coefficients for channels from the m-th UE to the k-th MEC server, respectively [6]. Then, we introduce $\rho_{m,k,l} = \mathbf{g}^T_{mk}\mathbf{g}^*_{lk}/\|\mathbf{g}_{lk}\|$.

Hence, the achievable transmission rate at the k-th MEC server according to the task of the m-th UE is given as [7]

$$R^{ul}_{mk}(\mathbf{p}, \boldsymbol{\pi}) = B\log_2\left(1 + \frac{\pi_{mk}p_m|\rho_{m,k,m}|^2}{\mathcal{I}_m(\mathbf{p}, \boldsymbol{\pi}) + \sigma^2_k}\right), \quad (1)$$

where p_m is the transmit power of the m-th UE, σ^2_k is the noise variance, and $\mathbf{p} = [p_m]^M_{m=1}$; $\mathcal{I}_m(\mathbf{p}, \boldsymbol{\pi}) = \sum_{l \in \mathcal{M}, l \neq m} \pi_{lk}p_l|\rho_{m,k,l}|^2$ represents the interference imposed on the AP.

The latency between the m-th UE and the k-th edge server for task offloading can be expressed as $T^{cm}_{mk}(\mathbf{p}, \boldsymbol{\pi}, \beta_{mk}) = \frac{\pi_{mk}\beta_{mk}D_m}{R^{ul}_{mk}(\mathbf{p},\boldsymbol{\pi})}$.

B. Computation model of physical and DT objects

1) Local processing: The task I_m at the m-th UE executes α_m portion of the task with the processing rate f_m. Let $C_m = \xi D_m$ (in cycles) denote the required computation resource, where ξ is the complexity of the task in cycles/bit. The estimated time required to execute the task locally is given by $\tilde{T}^{lc}_m(\alpha_m, f_m) = \frac{\alpha_m C_m}{f_m}$. Assuming that the deviation of the CPU processing frequency between the physical IoT and their DT counterparts can be acquired in advanced [1], the computing latency gap between the real value and DT estimation can be calculated by $\Delta T^{lc}_m(\alpha_m, f_m) = \frac{\alpha_m C_m \tilde{f}_m}{f_m\left(f_m - \tilde{f}_m\right)}$. Consequently, the actual time for local computing at the m-th UE can be expressed as $T^{lc}_m = \Delta T^{lc}_m + \tilde{T}^{lc}_m$.

2) Edge processing: The estimated latency of the k-th MEC server to execute task I_m is given by $\tilde{T}^{ed}_{mk}(\boldsymbol{\pi}_{mk}, \beta_{mk}, f_k) = \frac{\pi_{mk}\beta_{mk}C_m}{\tilde{f}_k}$. Then, the latency gap ΔT^{ed}_m between real value and DT estimation can be expressed as $\Delta T^{ed}_{mk}(\pi_{mk}, \beta_{mk}, f_k) = \frac{\pi_{mk}\beta_{mk}C_m\tilde{f}_k}{f_k\left(f_k - \tilde{f}_k\right)}$. As a result, the actual latency for executing at the edge DT can be expressed as $T^{ed}_{mk} = \Delta T^{ed}_{mk} + \tilde{T}^{ed}_{mk}$.

C. Total latency of the DT system

For the task I_m, the total DT latency in the system can be expressed as follows: $T^{tot}_m(\boldsymbol{\pi}, \alpha_m, \beta_{mk}, f_m, f_k, \mathbf{p}) = T^{lc}_m + \max_{k \in \mathcal{K}} T^{cm}_{mk} + \max_{k \in \mathcal{K}} T^{ed}_{mk} = \frac{\alpha_m C_m}{f_m - \tilde{f}_m} + \max_{k \in \mathcal{K}} \frac{\pi_{mk}\beta_{mk} + D_m}{R^{ul}_{mk}(\mathbf{p},\boldsymbol{\pi})} + \max_{k \in \mathcal{K}} \left(\frac{\pi_{mk}\beta_{mk}C_m}{f_k - \tilde{f}_k}\right)$.

D. Energy consumption model

The total energy consumption of the m-th UE with computation (E^{cp}_m) and transmission (E^{cm}_m) is given as $E^{tot}_m(\alpha_m, p_m, \boldsymbol{\pi}, f_m) = E^{cp}_m + E^{cm}_m = \alpha_m \frac{\theta}{2} C_m (f_m - \tilde{f}_m)^2 + \sum_{k \in \mathcal{K}} p_m \frac{\pi_{mk}\beta_{mk}D_m}{R^{ul}_{mk}(\mathbf{p},\boldsymbol{\pi})}$, where $\theta/2$ is a constant denoting the average switched capacitance and the average activity factor of the m-th UE [4].

E. Problem formulation

In this paper, our main objective is to minimise the total DT latency based on optimising the edge selection, offloading policies, transmit power, and estimated CPU frequency of the UEs and MEC servers. Hence, in the sequel, we will formulate the optimisation problem (2) with respect to the following constraints: i) the constraint (2b) represents the power constraint at the UEs with P_{max} denoting the maximum transmit power; ii) the constraint (2c) is maximum latency constraint for every incoming task; iii) the constraints (2d) and (2e) mean that each edge server can serve maximum of

M_{max} UEs; iv) the constraints (2f) and (2g) present value range and constraint of offloading factors; v) the constraints (2h) and (2i) are the minimum transmission rate requirement for uplink transmission from the UEs to the MEC servers and the maximum energy consumption requirement of the UEs; and vi) the constraints (2j) and (2k) reflect the computation resource limitations at the UEs and the MEC servers, respectively.

$$\begin{aligned}
&\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\pi},\mathbf{p},\mathbf{f}} \max_{\forall m\in\mathcal{M}} \left\{T_m^{tot}\left(\boldsymbol{\pi},\alpha_m,\beta_{mk},f_m,f_k,\mathbf{p}\right)\right\} && \text{(2a)}\\
&\quad \text{s.t. } p_m \le P_{max}, \forall m\in\mathcal{M}, && \text{(2b)}\\
&\qquad T_m^{tot}\left(\boldsymbol{\pi},\alpha_m,\beta_{mk},f_m,f_k,\mathbf{p}\right) \le T_m^{max}, && \text{(2c)}\\
&\qquad \pi_{mk}\in\{0,1\}, \forall m\in\mathcal{M},\forall k\in\mathcal{K}, && \text{(2d)}\\
&\qquad \sum_{m\in\mathcal{M}} \pi_{mk} \le M_{max}, \forall k\in\mathcal{K}, && \text{(2e)}\\
&\qquad \alpha_m\in[0,1], \beta_{mk}\in[0,1], \forall m\in\mathcal{M},\forall k\in\mathcal{K}, && \text{(2f)}\\
&\qquad \alpha_m + \sum_{k\in\mathcal{K}} \pi_{mk}\beta_{mk} = 1, \forall m\in\mathcal{M}, && \text{(2g)}\\
&\qquad R_{mk}^{ul}\left(\mathbf{p},\boldsymbol{\pi}\right) \ge \pi_{mk}R_{min}^{ul}, \forall m\in\mathcal{M},\forall k\in\mathcal{K}, && \text{(2h)}\\
&\qquad E_m^{tot}\left(\alpha_m,p_m,\boldsymbol{\pi},f_m\right) \le E_m^{max}, \forall m\in\mathcal{M}, && \text{(2i)}\\
&\qquad \alpha_m f_m \le f_m^{max}, \forall m\in\mathcal{M}, && \text{(2j)}\\
&\qquad \sum_{m\in\mathcal{M}} \pi_{mk}\beta_{mk}f_k \le f_k^{max}, \forall k\in\mathcal{K}, && \text{(2k)}
\end{aligned}$$

where $\boldsymbol{\alpha} = \{\alpha_m\}, \forall m\in\mathcal{M}$; $\boldsymbol{\beta} = \{\beta_{mk}\}, \forall m\in\mathcal{M}, \forall k\in\mathcal{K}$; $\boldsymbol{\pi} = \{\pi_{mk}\}, \forall m\in\mathcal{M}, \forall k\in\mathcal{K}$; $\mathbf{f} = \{f_m, f_k\}, \forall m\in\mathcal{M}, \forall k\in\mathcal{K}$.

However, the problem (2) is a non-convex problem with the non-convexity of (2a), (2c), (2h), (2i) which is difficult to solve. Since large-scale scenarios are considered, the complexity of problem (2) significantly increases with the large number of UEs and MEC servers. Therefore, we propose a distributed solution for solving the problem (2).

III. Distributed Solution for Optimal System Resource Allocation

We develop an iterative method that effectively solves problem (2). Specifically, first, we relax the integer variables in user association indicators ($\boldsymbol{\pi}$) into continuous variables. Then, we iteratively optimise the power allocation, offloading policies, and estimated processing rates of UEs and MEC servers.

A. Optimal power allocation with fixed $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\pi},\boldsymbol{f}$

Given $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\pi},\mathbf{f}$, the problem (2) can be reduced to

$$\begin{aligned}
&\min_{\mathbf{p}} \max_{\forall m\in\mathcal{M}} \left\{T_m^{tot}\left(\mathbf{p}\right)\right\} && \text{(3a)}\\
&\text{s.t. (2b), (2c), (2h), (2i).}
\end{aligned}$$

To solve problem (3), we use the logarithmic inequality given in [5], [8], which follows from the convexity of the function $f(x,y) = \log_2\Big(1+1/xy\Big)$ as

$$f(x,y) = \log_2(1+\frac{1}{xy}) \ge \hat{f}(x,y), \tag{4}$$

where, for $\forall x>0, \bar{x}>0, y>0, \bar{y}>0$, we have $\hat{f}(x,y) = \log_2\left(1+\frac{1}{\bar{x}\bar{y}}\right) + \frac{2}{(\bar{x}\bar{y}+1)} - \frac{x}{\bar{x}(\bar{x}\bar{y}+1)} - \frac{y}{\bar{y}(\bar{x}\bar{y}+1)}$. Let i denote the ith iteration and exploit $x = \frac{1}{\pi_{mk}p_m|\rho_{m,k,m}|^2}$, $y = \mathcal{I}_m(\mathbf{p})+\sigma_k^2$, $\bar{x} = x^{(i)} = \frac{1}{\pi_{mk}p_m^{(i)}|\rho_{m,k,m}|^2}$, and $\bar{y} = y^{(i)} = \mathcal{I}_m(\mathbf{p}^{(i)}) + \sigma_k^2$ for the approximation of the achievable transmission rate at the k-th MEC server in (1) as

$$R_{mk}^{ul}(\mathbf{p}) \ge \hat{R}_{mk}^{ul(i)}(\mathbf{p}), \forall m\in\mathcal{M}, \forall k\in\mathcal{K}, \tag{5}$$

where

$$\begin{aligned}
\hat{R}_{mk}^{ul(i)}(\mathbf{p}) = B\Big(&\log_2\Big(1+\frac{1}{\bar{x}\bar{y}}\Big) + \frac{2}{(\bar{x}\bar{y}+1)}\\
&-\frac{x}{\bar{x}(\bar{x}\bar{y}+1)} - \frac{y}{\bar{y}(\bar{x}\bar{y}+1)}\Big).
\end{aligned} \tag{6}$$

Hence, the constraint (2h) can be rewritten as

$$\hat{R}_{mk}^{ul(i)}(\mathbf{p}) \ge \pi_{mk}R_{min}^{ul}, \ \forall m\in\mathcal{M}, \forall k\in\mathcal{K}. \tag{7}$$

Next, by introducing the new variables $\mathbf{r} \triangleq \{r_{mk}\}$ ($\forall m\in\mathcal{M}, \forall k\in\mathcal{K}$) that satisfy $\frac{1}{R_{mk}^{ul}(\mathbf{p})} \le r_{mk}$, the objective function $T_m^{tot}(\mathbf{p})$ can be upper-bounded as $T_m^{tot}(\mathbf{p}) \le \hat{T}_m^{tot}(\mathbf{r}) = \frac{\alpha_m C_m}{f_m - \tilde{f}_m} + \max_{k\in\mathcal{K}}\left(\frac{\pi_{mk}\beta_{mk}C_m}{f_k-\tilde{f}_k}\right) + \max_{k\in\mathcal{K}}\{r_{mk}\pi_{mk}\beta_{mk}D_m\}$.

We can express (2c) and (2i) as

$$\begin{cases}
\hat{T}_m^{tot}(\mathbf{r}) \le T_m^{max}, & \text{(8a)}\\
\alpha_m\dfrac{\theta}{2}C_m f_m^2 + \displaystyle\sum_{k\in\mathcal{K}} p_m r_{mk}\pi_{mk}\beta_{mk}D_m \le E_m^{max}, & \text{(8b)}\\
\dfrac{1}{\hat{R}_{mk}^{ul(i)}(\mathbf{p})} \le r_{mk}, & \text{(8c)}\\
\forall m\in\mathcal{M}, \forall k\in\mathcal{K}. & \text{(8d)}
\end{cases}$$

Since the constraint (8b) is still non-convex, we apply the following inequality

$$x_2 y_2 \le \frac{1}{2}\left(\frac{\bar{y}_2}{\bar{x}_2}x_2^2 + \frac{\bar{x}_2}{\bar{y}_2}y_2^2\right), \tag{9}$$

with $x_2 = p_m$, $\bar{x}_2 = p_m^{(i)}$, $y_2 = r_{mk}$, $\bar{y}_2 = r_{mk}^{(i)}$, to iteratively express (8b) as

$$\begin{aligned}
&\sum_{k\in\mathcal{K}} \frac{1}{2}\left(\frac{r_{mk}^{(i)}}{p_m^{(i)}}p_m^2 + \frac{p_m^{(i)}}{r_{mk}^{(i)}}r_{mk}^2\right)\pi_{mk}\beta_{mk}D_m\\
&+\alpha_m\frac{\theta}{2}C_m f_m^2 \le E_m^{max}, \forall m\in\mathcal{M}, \forall k\in\mathcal{K}.
\end{aligned} \tag{10}$$

Consequently, problem (3) is equivalent to the following problem to generate a feasible point at the ith iteration:

$$\begin{aligned}
&\min_{\mathbf{p},\mathbf{r}} \max_{\forall m\in\mathcal{M}} \left\{\hat{T}_m^{tot}(\mathbf{r})\right\}, && \text{(11a)}\\
&\text{s.t. (2b), (8a), (7), (8c), (10).}
\end{aligned}$$

Hence, problem (11) is now a standard convex program and can be efficiently solved by convex optimisation solvers, e.g., CVX [9]. We propose a power allocation procedure for solving problem (11), as summarised in Algorithm 1.

B. Optimal edge selection with fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{p}, \boldsymbol{f}$

For fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{p}, \mathbf{f}$, problem (2) can be simplified as

$$\min_{\boldsymbol{\pi}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{tot}(\boldsymbol{\pi}) \right\} \tag{12a}$$

$$\text{s.t. } (2c), (2d), (2e), (2g), (2h), (2i), (2k).$$

Algorithm 1 : Optimal power allocation procedure for solving problem (11)

:

```
Input:
    Set $i = 0$, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{f}$ and initial point $\mathbf{p}^{(0)}$;
    Set the tolerance $\varepsilon = 10^{-3}$, the maximum iterations $I_{max} = 20$ to stop the algorithm;
Repeat
    Solve problem (11) for the feasible solution $(\mathbf{p}^{(i+1)})$;
    Set $i = i + 1$;
Until Convergence or $i > I_{max}$;
Output: Optimal power control coefficients $(\mathbf{p}^*)$.
```

As observed in subproblem (12), the objective function and the constraints (2c), (2h), (2i) are non-convex. To solve subproblem (12), first we use the logarithmic inequality (4) to obtain the approximation of the achievable transmission rate at the k-th MEC server as

$$R_{mk}^{ul}(\boldsymbol{\pi}) \geq \hat{R}_{mk}^{ul(i)}(\boldsymbol{\pi}), m \in \mathcal{M}, \tag{13}$$

where

$$\hat{R}_{mk}^{ul(i)}(\boldsymbol{\pi}) = B\Big(\log_2 \Big(1 + \frac{1}{\bar{x}_3 \bar{y}_3}\Big) + \frac{2}{(\bar{x}_3 \bar{y}_3 + 1)} - \frac{x_3}{\bar{x}_3(\bar{x}_3 \bar{y}_3 + 1)} - \frac{y_3}{\bar{y}_3(\bar{x}_3 \bar{y}_3 + 1)} \Big), \tag{14}$$

with $x_3 = \frac{1}{\pi_{mk} p_m |\rho_{m,k,m}|^2}$, $y_3 = \mathcal{I}_m(\boldsymbol{\pi}) + \sigma_k^2$, $\bar{x}_3 = x_3^{(i)} = \frac{1}{\pi_{mk}^{(i)} p_m |\rho_{m,k,m}|^2}$, and $\bar{y}_3 = y_3^{(i)} = \mathcal{I}_m(\boldsymbol{\pi}^{(i)}) + \sigma_k^2$. Hence, constraint (2h) is equivalently approximated as

$$\hat{R}_{mk}^{ul(i)}(\boldsymbol{\pi}) \geq \pi_{mk} R_{min}^{ul}. \tag{15}$$

Second, we introduce new variables $\tilde{\mathbf{r}} \triangleq \{\tilde{r}_{mk}\}$ ($\forall m \in \mathcal{M}, \forall k \in \mathcal{K}$) that satisfy $\frac{1}{R_{mk}^{ul}(\boldsymbol{\pi})} \leq \tilde{r}_{mk}$. By following the same inequality (9), we can express (2c) and (2i) as in the following constraints

$$\hat{T}_m^{tot}(\boldsymbol{\pi}, \tilde{\mathbf{r}}) = \frac{\alpha_m C_m}{f_m - \tilde{f}_m} + \max_{k \in \mathcal{K}} \left(\frac{\pi_{mk} \beta_{mk} C_m}{f_k - \tilde{f}_k} \right) + \max_{k \in \mathcal{K}} \left\{ \frac{1}{2} \Big(\frac{\tilde{r}_{mk}^{(i)}}{\pi_{mk}^{(i)}} \pi_{mk}^2 + \frac{\pi_{mk}^{(i)}}{\tilde{r}_{mk}^{(i)}} \tilde{r}_{mk}^2 \Big) \beta_{mk} D_m \right\} \leq T_m^{max} \tag{16}$$

$$\alpha_m \frac{\theta}{2} C_m f_m^2 + \sum_{k \in \mathcal{K}} \frac{1}{2} \Big(\frac{\tilde{r}_{mk}^{(i)}}{\pi_{mk}^{(i)}} \pi_{mk}^2 + \frac{\pi_{mk}^{(i)}}{\tilde{r}_{mk}^{(i)}} \tilde{r}_{mk}^2 \Big) p_m \beta_{mk} D_m \leq \ E_m^{max}, \tag{17}$$

$$\frac{1}{\hat{R}_{mk}^{ul(i)}(\boldsymbol{\pi})} \leq \ \tilde{r}_{mk}. \tag{18}$$

We note that from (16), the objective function $T_m^{tot}(\boldsymbol{\pi})$ can be upper-bounded as

$$T_m^{tot}(\boldsymbol{\pi}) \leq \hat{T}_m^{tot}(\boldsymbol{\pi}, \tilde{\mathbf{r}}). \tag{19}$$

Consequently, at the i-th iteration, we solve the following convex problem of (12):

$$\min_{\boldsymbol{\pi}, \tilde{\mathbf{r}}} \max_{\forall m \in \mathcal{M}} \left\{ \hat{T}_m^{tot}(\boldsymbol{\pi}, \tilde{\mathbf{r}}) \right\} \tag{20a}$$

$$\text{s.t. } (16), (2d), (2e), (2g), (15), (17), (18), (2k),$$

which is a convex program, and thus, can be efficiently solved by convex optimisation solvers. The procedure for solving problem (20) is similar to Algorithm 1. Therefore, we omit the details here.

C. Optimal offloading policies with fixed $\boldsymbol{\pi}, \mathbf{p}, \boldsymbol{f}$

For fixed $\boldsymbol{\pi}, \mathbf{p}, \mathbf{f}$, problem (2) can be rewritten as

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{tot}(\alpha_m, \beta_{mk}) \right\} \tag{21a}$$

$$\text{s.t. } (2c), (2f), (2g), (2i), (2j), (2k).$$

Obviously, problem (21) is a standard linear programming problem that can be solved by linear programming solvers.

D. Optimal estimated processing rates of UEs and MEC servers with fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{p}$

For fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{p}$, problem (2) can be simplified as

$$\min_{\mathbf{f}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{tot}(f_m, f_k) \right\} \tag{22a}$$

$$\text{s.t. } (2c), (2i), (2j), (2k),$$

since the constraints (2c), (2j), (2k) are convex with respect to $\mathbf{f}$. Hence, the problem (22) is convex with respect to $\mathbf{f}$ and can be efficiently solved by CVX.

Finally, based on the above analysis, we propose an iterative optimisation algorithm for efficiently solving the optimal resource allocation for UEs and MEC servers. The iterative algorithm is presented in Algorithm 2.

Algorithm 2 : Iterative optimisation algorithm for solving problem (2).

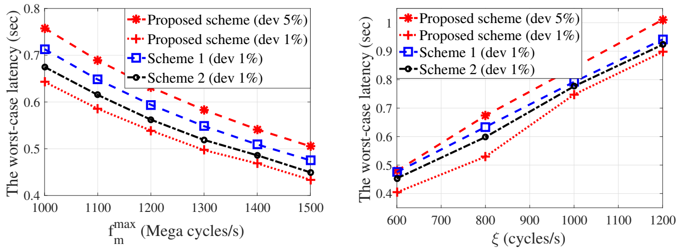
```
Input:
    Set $\kappa = 0$, initial point $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\pi}^{(0)}$, $\mathbf{p}^{(0)}$, $\mathbf{f}^{(0)}$;
    Set the tolerance $\varepsilon = 10^{-3}$, the maximum iterations $I_{max} = 20$ to stop the algorithm;
Repeat
    For $\boldsymbol{\alpha}^{(\kappa)}$, $\boldsymbol{\beta}^{(\kappa)}$, $\boldsymbol{\pi}^{(\kappa)}$, $\mathbf{f}^{(\kappa)}$, solve problem (11) for optimal power control coefficients $(\mathbf{p}^{(\kappa+1)})$;
    For $\boldsymbol{\alpha}^{(\kappa)}$, $\boldsymbol{\beta}^{(\kappa)}$, $\mathbf{p}^{(\kappa)}$, $\mathbf{f}^{(\kappa)}$, solve problem (12) for optimal edge selection $(\boldsymbol{\pi}^{(\kappa+1)})$;
    For $\boldsymbol{\pi}^{(\kappa)}$, $\mathbf{p}^{(\kappa)}$, $\mathbf{f}^{(\kappa)}$, solve problem (21) for optimal offloading policies $(\boldsymbol{\alpha}^{(\kappa+1)}, \boldsymbol{\beta}^{(\kappa+1)})$;
    For $\boldsymbol{\alpha}^{(\kappa)}$, $\boldsymbol{\beta}^{(\kappa)}$, $\boldsymbol{\pi}^{(\kappa)}$, $\mathbf{p}^{(\kappa)}$, solve problem (22) for optimal estimated processing rates $(\mathbf{f}^{(\kappa+1)})$;
    Set $\kappa = \kappa + 1$;
Until Convergence or $\kappa > I_{max}$;
Output: Optimal resource allocation $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$, $\boldsymbol{\pi}^*$, $\mathbf{p}^*$, $\mathbf{f}^*$.
```

Finally, assume that each MEC server selects M_{max} UEs to serve based on the M_{max} largest continuous variables in the user association indicators. Let $\mathcal{S}_k$ denote the set of M_{max} UEs served by the k-th MEC server. We convert the corresponding user association indicators to integer values $\pi_{mk} = 1$, $m \in \mathcal{S}_k$.

IV. Numerical results

We consider a MEC network where UEs are distributed randomly in a 100 $m \times 100$ m area with $M = [5, 8]$ UEs and $K = 2$ MEC servers. Each AP is equipped with $L = 4$ antennas. The channel path loss between the UEs and APs is modelled as $PL = 140.7 + 36.7 \log_{10} d$ (dB), with d as the geographical distance. The system bandwidth is set to $B = 10$ MHz, and the maximum transmit power of an UE is 30 dBm. The noise power density is set to $\sigma^2 = -174$ dBm/Hz. The maximum CPU cycle frequency of the UEs and the MEC servers is set to $f_m^{max} = 1.0$ Giga cycles/s and $f_k^{max} = 20$ Giga cycles/s, respectively, [3], [7]. The input data size at the m-th UE is set to $D_m = 100$ kB [2]. The complexity of the task is $\xi = [600, 1200]$ cycles/bit. The minimum data rate is $R_{min}^{ul} = 0.1$ Mbps and the maximum energy consumption is $E_m^{max} = 1.5$ Joule. The maximum latency constraint is $T_m^{max} = 1$ second. Each edge server can serve up to $M_{max} = 3$ UEs. The effective capacitance coefficient is $\theta_m = 10^{-24}$ $Watt.s^3/cycle^3$ [3], [4]. To indicate the advantage of our proposed method, we compare our approaches with conventional methods as follows:

- Our proposed scheme: combining optimal power allocation, optimal edge selection, optimal offloading policy, and optimal frequency/processing rate allocation.
- Scheme 1: without considering the optimal edge selection (i.e., only nearby selection). For edge selection, UEs always offload their computing task to the MEC server with minimum path loss.
- Scheme 2: without optimal offloading policy (i.e., equal offloading policy). For the offloading policy, UEs assign equal offloading factors to the associated MEC servers.



(a) Latency versus f_m^{max}

(b) Latency versus ξ

Fig. 1: The worst-case end-to-end latency performance with deviation of 1% and 5% versus (a) values of f_m^{max}, and (b) different values of ξ, with $K = 2$, $M = 6$.

In Fig. 1a, we evaluate the worst-case latency with respect to a range of f_m^{max} at $K = 2$, $M = 6$, $\xi = 900$ cycles/bit. As can be seen from the figure, the worst-case latency goes down with the increase of the computation resource limitations at the UEs. This is due to the fact that the larger the local computation resource limitations, the higher the capability to process the tasks locally instead of task offloading to MEC servers, and therefore, the higher the chance to reduce the latency especially with optimal allocation of network resource. Furthermore, we also investigate the impact of the deviation between the assigned CPU frequency in the real UEs/MEC servers and its DT by varying the different deviations of 1% and 5%. Specifically, for a fixed deviation of 1%, the average gain of the latency obtained with the proposed scheme is about 10% and 5% when compared with Scheme 1 and Scheme 2, respectively. It is also noticed that the latency performance decreased with the increase of the deviations. Fig. 1b shows how the worst-case latency changes with different values of ξ with $K = 2$, $M = 6$. We can observe that given the particular computation resource at the MEC servers and UEs, the worst-case latency is clearly an increasing function with task complexity, where part of the computing tasks at the UEs should be offloaded to the MEC servers in order to guarantee performance. Therefore, by making optimal offloading policies, the proposed scheme always provides a better performance than the benchmark schemes in terms of the worst-case latency. For instance, when ξ increases from 600 to 1200 cycles/s, the worst-case latency with the proposed method increases from some 0.4 to 0.9 seconds while with Scheme 2, the worst-case latency rises from approximately 0.45 to 0.95 seconds.

V. Conclusion

In this paper, we have proposed a novel DT framework assisting the task offloading of IoT devices for industrial IoTs networks with MEC. We then have solved the highly non-convex optimisation problem by minimising the end-to-end latency of the considered systems with respect to the transmission power, user association, task offloading, and CPU processing frequency. We have demonstrated that our proposed scheme outperforms the benchmark schemes.

References

[1] 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, pp. 12 240–12 251, Oct. 2020.

[2] Y. Zhou *et al.*, "Offloading optimization for low-latency secure mobile edge computing systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 480–484, April 2020.

[3] 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.*, pp. 1–1, 2021.

[4] W. Sun, P. Wang, N. Xu, G. Wang, and Y. Zhang, "Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles," *IEEE Internet Things J.*, pp. 1–1, 2021.

[5] L. D. Nguyen *et al.*, "Downlink beamforming for energy-efficient heterogeneous networks with massive MIMO and small cells," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3386–3400, May 2018.

[6] M.-H. T. Nguyen *et al.*, "Spectrum-sharing uav-assisted mission-critical communication: Learning-aided real-time optimisation," *IEEE Access*, vol. 9, pp. 11 622–11 632, 2021.

[7] M. Merluzzi *et al.*, "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, 2020.

[8] L. D. Nguyen *et al.*, "Multi-user regularized zero-forcing beamforming," *IEEE Trans. Signal Process.*, vol. 67, no. 11, pp. 2839 – 2853, 2019.

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