This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

Task Dependency Aware Optimal Resource Allocation for URLLC Edge Network: A Digital Twin Approach Using Finite Blocklength

Muhammad Awais, *Graduate Student Member, IEEE,* Haris Pervaiz, *Member, IEEE,* Qiang Ni, *Senior Member, IEEE,* and Wenjuan Yu, *Member, IEEE,*

***Abstract*—Next-generation wireless networks envision ubiquitous access and computational capabilities by seamlessly integrating aerial and terrestrial networks. Digital twin (DT) technology emerges as a proactive and cost-effective approach for resource-limited networks. Mobile edge computing (MEC) is pivotal in facilitating mobile offloading, particularly under the demanding constraints of ultra-reliable and low-latency communication (URLLC). This study proposes an advanced bisection sampling-based stochastic solution enhancement (BSSE) algorithm to minimize the system's overall energy-time cost by jointly optimizing task offloading and resource allocation strategies. The formulated problem is a mixed-integer nonlinear programming problem due to its inherently combinatorial linkage with task-offloading decisions and strong correlation with resource allocation. The proposed algorithm operates iteratively through the following steps: 1) narrowing the search space through a one-climb policy, 2) developing a closed-form solution for optimal CPU frequency and transmit power, and 3) implementing randomized task offloading, which updates it in the direction of reducing objective value. The scalability of the proposed algorithm is also analyzed for a two-device model, which is subsequently extended to multiple devices. Comparative analysis against benchmark schemes reveals that our approach reduces total energy-time cost by 15.35% to 33.12% when weighting parameter $\partial_{k_2}^{\lambda}$ is increased from 0.1 to 0.3, respectively.**

***Index Terms*—URLLC, Task Dependency Aware, Edge Network, Resource Allocation, and Digital Twins.**

I. Introduction

RECENT advancements in telecommunications and modern computing infrastructures have facilitated the provision of computation-intensive and time-sensitive applications [1]. The exponential increase in mobile device usage has led to the emergence of ultra-reliable and low-latency communication (URLLC), a technology striving to achieve a minimum delay of 1 millisecond and a high reliability of 99.999999% [2]. URLLC offers significant potential in supporting real-time operations, such as virtual reality, vehicular edge computing, and industrial automation [3]. Therefore, the integration of URLLC with other emerging technologies, including mobile edge computing (MEC), and digital twin (DT) is pivotal in accommodating a diverse range of mobile applications, encompassing industry 4.0, smart cities, and intelligent transportation systems [4]. However, its practical implementation causes intriguing and challenging concerns due to the complex relationship between reliability and end-to-end delay in next-generation wireless networks [5].

The rapid evolution of internet of things (IoTs) has facilitated the cost-effective connection of billions of wireless devices, introducing a new era of connectivity. However, IoT devices' limited battery life and computational power have emerged as a significant barrier, particularly in supporting computation-intensive applications within future-generation wireless networks [6]. These constraints are primarily attributed to concerns regarding production costs and stringent size restrictions. Aerial base stations assisted mobile edge computing (ABS-MEC) presents a promising solution to overcome these challenges, e.g., offering support for time-sensitive applications and computation-intensive services, encompassing a broad spectrum of IoT applications, i.e., from security to actuation and monitoring IoT systems. The primary goal is to optimize offloading decisions and minimize energy consumption (EC) of connected devices [7].

Mobile edge computing (MEC) represents an emerging, cost-effective paradigm that leverages computational and storage capabilities to support devices with limited resources [8]. Task offloading is the most significant feature of MEC, which enables resource-constrained IoT devices to offload their computation-intensive tasks to high-performance edge servers (ESs), either binary or partially. In binary offloading, each task is processed locally or offloaded to the ES. In contrast, each task is partitioned and executed locally and at the ES [9]. Our research primarily focuses on binary offloading within the context of finite blocklength (FBL), frequently employed in IoT systems to process tasks that cannot be partitioned. This approach is instrumental in fulfilling the escalating quality of service demands in edge networks, particularly in the context of resource allocation [10].

An innovative technology that has recently gained substantial prominence is the DT [11]. This innovative technology creates a virtual replica of physical systems, enabling the simulation of optimal solutions before real-time implementation. Its application significantly enhances system performance while minimizing downtime [12]. Consequently, various studies have extensively investigated and analyzed the fusion of DT technology with the URLLC edge network based on MEC to improve the performance and reliability of communication

Muhammad Awais, Qiang Ni, and Wenjuan Yu are with the School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, UK. e-mails: {m.awais11, h.b.pervaiz, w.yu8, q.ni}@lancaster.ac.uk.

Haris Pervaiz is with the School of Computer Science and Electronic Engineering (CSEE), University of Essex, UK. e-mail: haris.pervaiz@essex.ac.uk.

Manuscript received January 25, 2024; revised May 28, 2024.

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

systems [13]. In [14], the authors investigate the challenges of task offloading on the ES, incorporating the concepts of DT, blockchain, and channel state information (CSI) to boost network performance. The study presented in [15] proposed an ABS-based DT-assisted offloading solution, which jointly optimizes offloading factors, transmit power, and processing rate to minimize the overall delay. An iterative algorithm is introduced in [16], focusing on optimizing computing and communication parameters within a DT-based edge network. This algorithm aims explicitly to reduce latency for industrial IoTs operating under URLLC constraints.

A lower-bound methodology for maximizing data rate using massive multiple input multiple outputs (MIMO) is proposed for uplink URLLC, as detailed in [17]. This optimization is achieved by jointly fine-tuning the payload and pilot transmission strategies for zero-facing and maximum ratio combining designs. Reference [18] addresses the challenge of resource optimization in URLLC, employing a game theoretic approach to enhance the offloading factor within a multi-agent edge network [19]. A distributed solution to optimize the average response time for computational offloading is presented in [20]. The study is extended to a distributed framework that solves the NP-hard energy efficiency problem, leveraging parallel processing [21].

Channel characteristics are also crucial in the next generation of wireless networks. Therefore, it is not always recommended to completely offload computational tasks to the ES because it may lead the network towards a low offloading data rate due to the possibility of deep fading [22]. An illustrative example is an adaptive search algorithm, which minimizes EC through joint optimization of offloading factors, resource allocation, and user association [23]. Similarly, the authors in [24] investigate resource allocation and task offloading from an economic perspective to enhance computing efficiency. The research in [25] focuses on developing optimal binary offloading policies for single-user tasks, later extended to multi-user scenarios in subsequent studies [26]. This concept is further applied to multiple independent tasks, in the system where a single user offloads its task to different ESs [27], and multiple users offload their tasks to a single ES [28].

Tasks executed by different IoT devices usually correlate [29]- [30]. This interdependence significantly influences decisions related to offloading and resource allocation in an integrated aerial terrestrial edge network [31]–[33]. For instance, in poor channel conditions, a device might need to offload its tasks to an ABS-assisted edge server urgently required by another device. This exchange of computation outcomes incurs extra energy and time. Due to this strong coupling and combinatorial nature of the problem, identifying an optimal solution is challenging. It is noted that previous research has explored a variety of approaches to similar issues [15]–[17]. However, this work addresses the context of DT-aided edge computing for URLLC. In this domain, the task dependencies among interconnected wireless devices have not been sufficiently explored [31]–[35]. To the best of our knowledge, the current research initiates a pioneering examination of task dependency among devices within the context of DT-aided edge computing for URLLC. The key contributions of this work are outlined as follows:

- We have formalized a mixed-integer non-linear programming (MINLP) problem within a DT-enabled integrated aerial-terrestrial network, taking into account the interdependencies among tasks in a novel manner. Owing to its inherently combinatorial linkage with task-offloading decisions and strong correlation with resource allocation, this problem poses significant computational challenges. Therefore, we introduce an enhanced bisection sampling-based stochastic solution enhancement (BSSE) algorithm that aims to minimize the system's energy-time cost iteratively, offering a solution that closely matches the performance of the most effective existing scheme.
- To efficiently narrow the search space, we have followed 'one-climb policy', where a device offloads its data to the edge server at the optimum time only once. The proposed algorithm is specifically tailored to jointly optimize transmit power, CPU frequency, and the task offloading policy; thereby minimizing the weighted sum of the devices' energy consumption and task execution time. We derived a closed-form solution for calculating the optimal CPU frequency and transmit power for given offloading decisions. Then, an inequality condition is formulated to manage dependent tasks efficiently. The proposed algorithm commences with a random task offloading configuration and iteratively updates it to reduce the system's energy-time cost.
- The scalability of the proposed model is analyzed by varying the number of IoT devices with the sequential number of tasks, i.e., from a simplified two-device framework to multiple devices, incorporating different intermediate tasks. The trade-off between the system's energy-time cost is also analyzed to validate the effectiveness of our approach.
- Although the proposed algorithm can manage a diverse range of tasks, the computational complexity of our proposed approach is significantly lower than benchmark schemes. We have also compared our BSSE approach against three sophisticated benchmark schemes: the bisection algorithm [29], the one-climb policy-based Gibbs sampling algorithm [29], and the exhaustive search algorithm. A comparative analysis with the bisection algorithm reveals that our approach reduces the total energy-time cost by 15.35% to 33.12% when the weighting parameter $\partial_{k_2}^{\lambda}$ is increased from 0.1 to 0.3, respectively.

The remaining paper is organized as: Section II contains the system model. Section III provides the problem formulation. Section IV explains the proposed solution, later extended to a multi-device scenario in Section V. Simulation results are given in Section VI. The paper is concluded in Section VII.

II. System Model

A. Implementation of DT in Edge Network for URLLC

Fig. 1 illustrates a DT-enabled edge computing network including two layers. The physical layer includes IoT devices and ABSs acting as access points (capable of working as ES, i.e., executing back-haul processing to mitigate the constraints

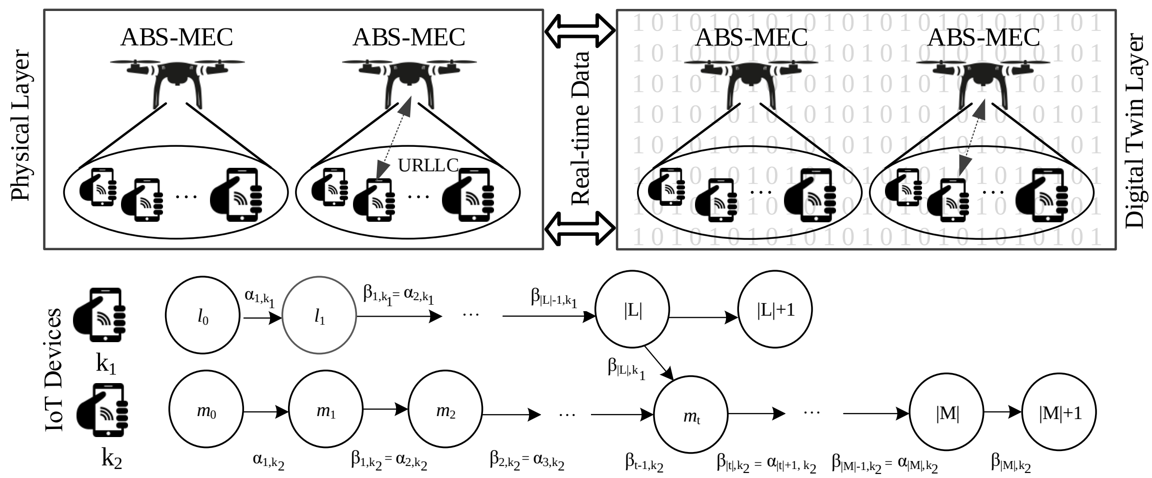

Fig. 1. Digital twin-enabled integrated aerial-terrestrial edge network denoting a two-device framework with an intermediate task node m_t

of limited storage and computing resources). The devices are connected via URLLC links to ensure stringent high-reliability and low-latency communication requirements in mission-critical applications. Each access point is assumed to be integrated with a multi-processor architecture with a fixed service rate. This configuration allows every access point to process a predetermined number of tasks simultaneously. The DT layer is a virtual replica of the physical layer, facilitating real-time monitoring of the physical system's operations.

B. Architecture of MEC based on URLLC

We consider a set of ABSs denoted by $a \in \mathcal{A} = \{a_1, a_2, ..., |\mathrm{A}|\}$. Each ABS serves a distinct non-overlapping region within its coverage area. It is assumed that each ABS has completed its deployment and networking planning beforehand. Within these coverage areas, we consider different numbers of IoT devices represented by the set $k \in \mathcal{K} = \{k_1, k_2, ..., k_j, ..., |\mathrm{K}|\}$. A binary variable is introduced to represent the connection between the device and ABS, i.e.,

$$\pi_{k,a} = \begin{cases} 1, & \text{if there is an association between the } (a)\text{-th} \\ & \text{ABS and the } (k)\text{-th IoT device} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For easy understanding, the number of IoT devices within each serving ABS is limited to two, i.e., IoT devices k_1 and k_2, exhibiting limited mobility. We have extended the proposed model to accommodate multi-user scenarios in Section (V). The sequential order of tasks to be performed by each respective device is denoted by the set $l \in \mathcal{L} = \{l_0, l_1, l_2, ..., |\mathrm{L}|, |\mathrm{L}|{+}1\}$ and the set $m \in \mathcal{M} = \{m_0, m_1, m_2, ..., |\mathrm{M}|, |\mathrm{M}|{+}1\}$. Two auxiliary nodes l_0 and m_0 are allocated as the starting points for the tasks assigned to each IoT device. Meanwhile, additional two nodes $|\mathrm{L}|{+}1$ and $|\mathrm{M}|{+}1$ are designated as termination points for the tasks assigned to each IoT device k_1 and k_2, respectively. We assume a framework of task dependency among IoT devices, where the intermediary task $t \in \{t_1, t_2, ..., |\mathrm{M}|\}$ of IoT device k_2 requires the output from the final task of device k_1. All tasks must be started and completed on the same device, e.g., the IoT device must execute two additional tasks locally. However, the remaining $|\mathrm{L}{+}\mathrm{M}|$ tasks may be processed locally or remotely.

It is reasonable to consider that the network under consideration is resource-constrained with limited bandwidth (W). The bandwidth is equally distributed among orthogonal subcarriers, represented by the set $s \in \mathcal{S} = \{s_1, s_2, ..., |\mathrm{S}|\}$, where $W = \sum_{s \in \mathcal{S}} w_s$. We define a binary variable for subcarrier allocation, where $\phi_{s,k} = 1$ indicates successful subcarrier allocation; otherwise, $\phi_{s,k} = 0$. In numerous power-efficient IoT structures, the rate required to offload the task is often minimal and generally necessitates only a narrow bandwidth; e.g., a narrow-band IoT system utilizing a 10 MHz bandwidth can accommodate over fifty IoT devices through orthogonal transmission. Consequently, every device is assigned an orthogonal subcarrier, having equal bandwidth. Our assumption to offload the task to an ES only once minimizes the probability of offloading all the data simultaneously.

C. Task Offloading Model in Edge Network

We represent the task i originating from the k^{th} device as a tuple $T_{i,k} = (\delta_{i,k}, \alpha_{i,k}, \beta_{i,k})$, where $i \in (l, m)$ and $k \in (k_1, k_2)$. In this context, $\delta_{i,k}$ specifies the computing resource requirement (cycles) necessary to execute the task, $\alpha_{i,k}$ indicates the input size of the task, and $\beta_{i,k}$ represents the output size of the task (bits). The computing resource requirement for auxiliary nodes corresponding to each IoT device is zero. Additionally for the given device k_1, the input of task l is equal to the output of the preceding task $l-1$, i.e., $\alpha_{l,k_1} = \beta_{l-1,k_1}$. For IoT device k_2, it is assumed that

$$\alpha_{m,k_2} = \begin{cases} \beta_{m-1,k_2} + \beta_{|L|,k_1} & \text{if } m = t, \\ \beta_{m-1,k_2} & \text{Otherwise.} \end{cases} \tag{2}$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

It is important to mention that the initial node is characterized by an input value of $\alpha_{i,k} = 0$, and the final node by an output value of $\beta_{i,k} = 0, \forall k \in \mathcal{K}$. To determine the feasibility of task offloading, we introduce a computational offloading decision variable, where $\varphi_{i,k} = 1$ means task i is processed at the edge; otherwise, $\varphi_{i,k} = 0$. Our analysis operates under non-causal channel information, implying that the access point has full access to the CSI while downloading or offloading tasks. These assumptions are critical in optimal offloading decisions.

D. DT Model

The DT of the ABS-assisted MEC within the URLLC edge network is articulated as follows [16].

$$DT = \{(K, \tilde{K}), (A, \tilde{A})\}. \tag{3}$$

The replica of the physical system is defined as $\tilde{K}$ and $\tilde{A}$. The following equation represents the DT for the k^{th} device.

$$DT_{i,k} = \left(f^{lo}_{i,k}, \hat{f}^{lo}_{i,k}\right). \tag{4}$$

The actual processing rate of the DT layer replicates the behavior of the physical IoT device is denoted by $f^{lo}_{i,k}$. Any deviation from the performance of the corresponding physical device is given by $\hat{f}^{lo}_{i,k}$ [34]. The DT model for the a^{th} ABS is articulated as follows.

$$DT_a = \left(f^{es}_a, \hat{f}^{es}_a\right). \tag{5}$$

The actual processing rate at which the physical ABS distributes the computing power of ES is denoted by f^{es}_a, and any deviation from the performance of the corresponding physical device is indicated by $\hat{f}^{es}_a$. It helps to minimize the processing latency gap at the DT layer by facilitating the adjustment of computing resource allocation. After gathering real-time data from the physical system, the digital services within the DT layer perform visualization and analysis to streamline and optimize decision-making, thereby enhancing system performance.

E. Communication Model

When device k offloads its task i to the access point a, the signal to interference plus noise ratio (SINR) is determined by the expression $\varrho^{es,s}_{i,k} = \left(\frac{\pi_{k,a}\phi_{s,k}\varphi_{i,k}p^s_{i,k}\|h^s_{i,k}\|^2}{\bar{\sigma}^2+\sigma_{i,k}}\right)$, where $p^s_{i,k}$ denotes the transmission power required for offloading task i by device k at subcarrier s, $\bar{\sigma}^2$ represents the spectral noise density, and $\sigma_{i,k}$ indicates the interference power caused by the neighbor IoT devices. The wireless channel between the k^{th} device and a^{th} ABS on the given subcarrier is represented by $h^s_{i,k}$ and can be modeled as $h^s_{i,k} = \sqrt{g^s_{i,k}}\tilde{h}^s_{i,k}$, where $\tilde{h}^s_{i,k}$ describes the small-scale fading with zero mean and uniform variance and $g^s_{i,k} = PL_{a,k} + \eta^{LoS}\rho^{LoS}_{a,k} + \eta^{NLoS}\rho^{NLoS}_{a,k}$ is the large-scale channel co-efficient, where ρ^{LoS} and ρ^{NLoS} are the additional loss for the line of sight (LoS) and non-light of sight (NLos) respectively. The pathloss between the given ABS and IoT device is computed by $PL_{a,k} = 10\log\left(\frac{4\pi f_c d_{a,k}}{c}\right)^2$, where $d_{a,k}$ is the Euclidean distance between the ABS a and (a,k)-th IoT device, f_c is the carrier frequency, and c is the speed of light. The probability for LoS is computed by $\rho^{LoS}_{a,k} = \frac{1}{1+\exp\left[-b\left(\arctan\left(\frac{h_a}{d_{a,k}}\right)-\nu\right)\right]}$ at height h_a, and fixed environmental factors b and ν. The probability for NLoS is given by $\rho^{NLoS}_{a,k} = 1 - \rho^{LoS}_{a,k}$ [35]. If the device k offloads the task i to the ES, then the uplink data rate is given by

$$r^{es,s}_{i,k} = w_s \log_2\left(1 + \varrho^{es,s}_{i,k}\right) - w_s\sqrt{\frac{v^s_{i,k}}{b_{i,k}}}\frac{Q^{-1}(\epsilon_{i,k})}{\ln 2}, \quad \text{where, } i \in \{l,m\}, k \in \{k_1,k_2\}. \tag{6}$$

The bandwidth allocated to the given subcarrier s is denoted by w_s, the channel dispersion is computed using $v^s_{i,k} = 1 - \left(1 + \varrho^{es,s}_{i,k}\right)^{-2}$, and the blocklength is denoted by variable $b_{i,k}$. The Gaussian Q-function is defined as $Q(x) = \frac{1}{2\pi}\int_x^\infty \exp(-\frac{t^2}{2})\, dt$ [36]. The transmission time required for offloading the task i from device k to the access point is expressed as $\eta^{es,s}_{i,k} = \frac{\beta_{i-1,k}}{r^{es,s}_{i,k}}$, where $i \in \{l,m\}, k \in \{k_1,k_2\}$. Assuming task i for device k is downloaded from the access point, the signal-to-noise ratio (SNR) is given by $\varrho^{dl,s}_{i,k} = \left(\frac{\pi_{k,a}\phi_{s,k}\varphi_{i,k}p_a\|h^s_{i,k}\|^2}{\bar{\sigma}^2+\sigma_{i,k}}\right)$, where p_a denotes the transmission power of the access point. The downlink data rate of task i for device k from the ES is given by

$$r^{dl,s}_{i,k} = w_s \log_2\left(1 + \varrho^{dl,s}_{i,k}\right) - w_s\sqrt{\frac{v^s_{i,k}}{b_{i,k}}}\frac{Q^{-1}(\epsilon_{i,k})}{\ln 2}, \quad \text{where, } i \in \{l,m\}, k \in \{k_1,k_2\}. \tag{7}$$

The channel dispersion for this link is computed by $v^s_{i,k} = 1 - (1 + \varrho^{dl,s}_{i,k})^{-2}$, and the time required for the downlink transmission is computed as $\eta^{dl,s}_{i,k} = \frac{\beta_{i-1,k}}{r^{dl,s}_{i,k}}$, where $i \in \{l,m\}, k \in \{k_1,k_2\}$.

F. Computational Model

1) Local Computing: The time required for local computation of task i on device k using the actual processing rate is expressed as [16].

$$\eta^{lo,s}_{i,k} = \frac{\delta_{i,k}}{f^{lo}_{i,k}}, i \in \{l,m\}, k \in \{k_1,k_2\}. \tag{8}$$

Assuming the deviation between the physical value and its DT representation is predetermined [16], the latency gap for task i on device k is given by

$$\Delta\eta^{lo,s}_{i,k} = \frac{\delta_{i,k}\hat{f}^{lo}_k}{f^{lo}_k\left(f^{lo}_{i,k} - \hat{f}^{lo}_{i,k}\right)}, i \in \{l,m\}, k \in \{k_1,k_2\}. \tag{9}$$

Consequently, the total local computation time for task i on device k is computed as

$$\eta^{lc,s}_{i,k} = \eta^{lo,s}_{i,k} + \Delta\eta^{lo,s}_{i,k}, i \in \{l,m\}, k \in \{k_1,k_2\}. \tag{10}$$

2) Edge Computing: The anticipated execution time for task i on device k when processed at the ES is given by

$$\tilde{\eta}^{es,s}_{i,k} = \frac{\delta_{i,k}}{f^{es}_{i,k}}, i \in \{l,m\}, k \in \{k_1,k_2\}, \tag{11}$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

where $f_{i,k}^{es}$ denotes the fixed frequency of the CPU at the ES. The latency gap between the real value and its DT is given by

$$\Delta\eta_{i,k}^{es,s} = \frac{\delta_{i,k}\hat{f}_{i,k}^{es}}{f_{i,k}^{es}\left(f_{i,k}^{es} - \hat{f}_{i,k}^{es}\right)}, i \in \{l,m\}, k \in \{k_1,k_2\}. \quad (12)$$

G. Latency Model

The end-to-end latency within the network is given by

$$\eta_{i,k}^{tot,s} = \eta_{i,k}^{lc,s} + \eta_{i,k}^{es,s} + \overbrace{\tilde{\eta}_{i,k}^{es,s} + \Delta\eta_{i,k}^{es,s}}^{\eta_{i,k}^{ec,s}}. \quad (13)$$

H. Energy Consumption Model

The local EC required for computing task i is calculated as follows [29].

$$\xi_{i,k}^{lo,s} = \mu\frac{\left(\delta_{i,k}\right)^3}{\left(\eta_{i,k}^{lo,s}\right)^2} \approx \mu\delta_{i,k}\left(f_{i,k}^{lo} - \hat{f}_{i,k}^{lo}\right)^2, \quad \text{where, } i \in \{l,m\}, k \in \{k_1,k_2\}, \quad (14)$$

where μ represents the switched capacitance coefficient of the IoT device [16]. Let us define $f(x) = \sigma^2\left(2^{\left(\frac{x}{w_s}\right)} - 1\right)$, we get using (6) as

$$p_{i,k}^{s} = \frac{1}{\|h_{i,k}^{s}\|^2} f\left(\frac{\beta_{i-1,k}}{\eta_{i,k}^{es,s}}\right), i \in \{l,m\}, k \in \{k_1,k_2\}. \quad (15)$$

Equation (15) shows the transmit power depends on channel conditions and data requirements. For example, an increase in the distance leads to a higher pathloss; therefore, it requires more transmission power to offload tasks effectively or to maintain a constant level of received signal power. The transmission energy required to offload task i at the access point is computed as

$$\xi_{i,k}^{es,s} = p_{i,k}^{s}\eta_{i,k}^{es,s} = \frac{\eta_{i,k}^{es,s}}{\|h_{i,k}^{s}\|^2 f\left(\frac{\beta_{i-1,k}}{\eta_{i,k}^{es,s}}\right)} \quad \text{where, } i \in \{l,m\}, k \in \{k_1,k_2\}. \quad (16)$$

The total EC is computed as

$$\xi_{i,k}^{tot,s} = \xi_{i,k}^{lo,s} + \xi_{i,k}^{es,s}. \quad (17)$$

I. Task Dependency Use Cases

The task dependency between two IoT devices can be one of the following four cases.

- Case I: When both IoT devices k_1 and k_2 offload the tasks $|\mathrm{L}|$ and t at the edge, i.e., $\varphi_{|\mathrm{L}|,k_1} = 1$ and $\varphi_{t,k_2} = 1$, then there is no need to offload the task or download the task.
- Case II: When device k_1 offloads task $|\mathrm{L}|$ at the edge and device k_2 computes task t locally, i.e., $\varphi_{|\mathrm{L}|,k_1} = 1$ and $\varphi_{t,k_2} = 0$, the resultant data from task $|\mathrm{L}|$ is transmitted to the IoT device k_2 after the completion of its computational processing at the edge node.
- Case III: When devices k_2 offloads task t at the edge and device k_1 computes task $|\mathrm{L}|$ locally, i.e., $\varphi_{t,k_2} = 1$ and $\varphi_{|\mathrm{L}|,k_1} = 0$, the IoT device k_1 needs to offload the results before the computation of task t at the edge node.
- Case IV: When both IoT devices k_1 and k_2 execute tasks $|\mathrm{L}|$ and t to locally, i.e., $\varphi_{|\mathrm{L}|,k_1} = 0$ and $\varphi_{t,k_2} = 0$, device k_1 first offloads its output to the edge (EDGE acting as a RELAY), then edge forwards it information to IoT device k_2. Therefore, the offloading transmission time is computed as $\eta_{|\mathrm{L}|+1,k_1}^{es,s} = \frac{\beta_{|\mathrm{L}|,k_1}}{r_{|\mathrm{L}|+1,k_1}^{es,s}}$, where $r_{|\mathrm{L}|+1,k_1}^{es,s}$ denotes the corresponding offloading data rate. The offloading EC is given by $\xi_{|\mathrm{L}|+1,k_1}^{es,s} = \left(p_{|\mathrm{L}|+1,k_1}^{s} \times \eta_{|\mathrm{L}|+1,k_1}^{es,s}\right)$, where $p_{|\mathrm{L}|+1,k_1}^{s}$ represents the offloading transmission power. The downlink transmission time is computed as $\eta_{t',k_2}^{dl,s} = \frac{\beta_{|\mathrm{L}|,k_1}}{r_{t,k_2}^{dl,s}}$, where $r_{t,k_2}^{dl,s}$ is the corresponding downlink rate.

III. Problem Formulation

The computational time required by IoT device k_1 involves local processing time and at the ES, which is given by

$$\lambda_{k_1}^{tcom} = \sum_{l=1}^{|\mathrm{L}|}\Big[\overbrace{(1-\varphi_{l,k_1})\eta_{l,k_1}^{lo,s}}^{local} + \overbrace{\varphi_{l,k_1}\tilde{\eta}_{l,k_1}^{es,s}}^{edge}\Big]. \quad (18)$$

The total computational delay for IoT device k_1 on offloading or downloading the task to/from the ES is given by

$$\lambda_{k_1}^{trans} = \sum_{l=1}^{|\mathrm{L}|+1}\Big[\overbrace{\varphi_{l,k_1}(1-\varphi_{l-1,k_1})\eta_{l,k_1}^{es,s}}^{offloading} + \overbrace{(1-\varphi_{l,k_1})\varphi_{l-1,k_1}\eta_{l,k_1}^{dl,s}}^{downloading}\Big]. \quad (19)$$

Transmission delay is zero when both tasks are processed on the same IoT device, i.e., $\varphi_{l-1,k_1} = \varphi_{l,k_1}$. Conversely, a delay occurs during the offloading process if $\varphi_{l-1,k_1} = 0$ and $\varphi_{l,k_1} = 1$. Similarly, there is a downloading delay when $\varphi_{l-1,k_1} = 1$ and $\varphi_{l,k_1} = 0$. Therefore, the total execution time for device k_1 is computed as $\lambda_{k_1}^{tot} = \lambda_{k_1}^{tcom} + \lambda_{k_1}^{trans}$. The total EC of IoT device k_1 is the cumulative sum of the EC of $|\mathrm{L}|$ tasks and the EC required for offloading the output of the final result if executed locally. The total EC is expressed as follows.

$$\xi_{k_1} = \overbrace{\sum_{l=1}^{|\mathrm{L}|}\Big[(1-\varphi_{l,k_1})\xi_{l,k_1}^{lo,s} + \varphi_{l,k_1}(1-\varphi_{l-1,k_1})\xi_{l,k_1}^{es,s}\Big]}^{\text{EC of the } |\mathrm{L}| \text{ tasks}} + \overbrace{(1-\varphi_{|\mathrm{L}|,k_1})\xi_{|\mathrm{L}|+1,k_1}^{es,s}}^{\text{EC while offloading final result, if } \varphi_{|\mathrm{L}|,k_1}=0}. \quad (20)$$

It is worth mentioning that the energy required to offload a given task, denoted by $\xi_{l,k_1}^{es,s}$ occurs only when $\varphi_{l,k_1} = 1$ and $\varphi_{l-1,k_1} = 0$. Similarly, the total EC for IoT device k_2 is computed as follows.

$$\xi_{k_2} = \sum_{m=1}^{|\mathrm{M}|}\Big[(1-\varphi_{m,k_2})\xi_{m,k_2}^{lo,s} + \varphi_{m,k_2}(1-\varphi_{m-1,k_2})\xi_{m,k_2}^{es,s}\Big]. \quad (21)$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

$$\xi_{k_1}^{hold} = \overbrace{\sum_{l=1}^{|\mathrm{L}|}\Big[(1-\varphi_{l,k_1})\eta_{l,k_1}^{lo,s} + \varphi_{l,k_1}(\tilde{\eta}_{l,k_1}^{es,s} + \eta_{l,k_1}^{es,s}) + \varphi_{l-1,k_1}\eta_{l,k_1}^{dl,s} - \varphi_{l-1,k_1}\varphi_{l,k_1}(\eta_{l,k_1}^{dl,s} + \eta_{l,k_1}^{es,s})\Big]}^{\text{Total amount of time for } |\mathrm{L}| \text{ tasks}} + \overbrace{(1-\varphi_{|L|,k_1})\eta_{|\mathrm{L}|+1,k_1}^{es,s} + (1-\varphi_{t,k_2})\eta_{t',k_2}^{dl,s}}^{\text{Transmission time of the output of } |\mathrm{L}| \text{ tasks by IoT device } k_1}. \tag{22}$$

$$\xi_{k_2}^{\text{hold}} = \overbrace{\sum_{m=1}^{|\mathrm{t}|-1}\Big[(1-\varphi_{m,k_2})\eta_{m,k_2}^{\text{lo,s}} + \varphi_{m,k_2}(\tilde{\eta}_{m,k_2}^{\text{es,s}} + \eta_{m,k_2}^{\text{es,s}}) + \varphi_{m-1,k_2}\eta_{m,k_2}^{\text{dl,s}} - \varphi_{m-1,k_2}\varphi_{m,k_2}(\eta_{m,k_2}^{\text{dl,s}} + \eta_{m,k_2}^{\text{es,s}})\Big]}^{\text{Total execution time for first } |\mathrm{t}|-1 \text{ tasks}} + \overbrace{\varphi_{t,k_2}\eta_{t,k_2}^{\text{es,s}} + \varphi_{t-1,k_2}\eta_{t,k_2}^{\text{dl,s}} - \varphi_{t-1,k_2}\varphi_{t,k_2}(\eta_{t,k_2}^{\text{dl,s}} + \eta_{t,k_2}^{\text{es,s}})}^{\text{Transmission time to offload task } |\mathrm{t}|,\text{ i.e., } \varphi_{t-1,k_2}=0,\ \varphi_{t,k_2}=1 \text{ or to download output for } |\mathrm{t}|-1 \text{ task to device } k_2,\text{ i.e., } \varphi_{t-1,k_2}=1,\ \varphi_{t,k_2}=0}. \tag{23}$$

$$\begin{aligned}\xi_{k_2} = \xi^{\text{hold}} &+ \sum_{m=t}^{|\mathrm{M}|}\Big[(1-\varphi_{m,k_2})\eta_{m,k_2}^{\text{lo,s}} + \varphi_{m,k_2}(\tilde{\eta}_{m,k_2}^{\text{es,s}})\Big] \\ &+ \sum_{m=t+1}^{|\mathrm{M}|+1}\Big[\varphi_{m,k_2}\eta_{m,k_2}^{\text{es,s}} + \varphi_{m-1,k_2}\eta_{m,k_2}^{\text{dl,s}} - \varphi_{m-1,k_2}\varphi_{m,k_2}(\eta_{m,k_2}^{\text{dl,s}} + \eta_{m,k_2}^{\text{es,s}})\Big].\end{aligned} \tag{24}$$

For execution time for IoT device k_2, we consider the waiting time to reach the final output of the IoT device k_1 to IoT device k_2. It consists of the total execution time to compute $|\mathrm{L}|$ computational tasks from IoT device k_1 and transmission time to offload the final output of the device k_1 (refer to Fig. 1) is given in (22). The time required for IoT device k_2, until the output of the task $|\mathrm{t}|-1$ is ready is expressed in (23). Utilizing (22) and (23), the total time required before the t^{th} task of device k_2 is ready to execute is given by $\xi^{hold} = \max\{\xi_{k_1}^{hold}, \xi_{k_2}^{hold}\}$. The total execution time of device k_2 is formulated by adding ξ^{hold} and the time required to complete the tasks from t to $|\mathrm{M}|$ is given in (24).

This study aims to reduce the energy-time cost for URLLC edge networks. Such a reduction is achieved by optimizing the offloading policy and allocating the resources, e.g., transmission power to offload the task and CPU frequency. Let $0 \leq \partial_{k_1}^{\xi} \leq 1$ and $0 \leq \partial_{k_1}^{\lambda} \leq 1$ are the weighting factors for the EC and execution time of the given device k_1. In this context, if we have $\partial_{k_1}^{\xi} + \partial_{k_1}^{\lambda} = 1$ then $\partial_{k_1}^{\xi} = 1 - \partial_{k_1}^{\lambda}$. Following real-time requirements, each IoT device can select weights (higher or lower) to fulfill user-oriented needs. So, the energy-time cost for the device k_1 is given by $\zeta_{k_1} = \partial_{k_1}^{\xi}\xi_{k_1} + \partial_{k_1}^{\lambda}\lambda_{k_1}^{tot}$. Let $0 \leq \partial_{k_2}^{\xi} \leq 1$ and $0 \leq \partial_{k_2}^{\lambda} \leq 1$ are the weighting factors for the EC and execution time of the given device k_2. In this context, if we have $\partial_{k_2}^{\xi} + \partial_{k_2}^{\lambda} = 1$ then $\partial_{k_2}^{\xi} = 1 - \partial_{k_2}^{\lambda}$. The energy-time cost for the device k_2 is given by $\zeta_{k_2} = \partial_{k_2}^{\xi}\xi_{k_2} + \partial_{k_2}^{\lambda}\lambda_{k_2}^{tot}$. We assume $\boldsymbol{\varphi} \triangleq \{\varphi_{i,k}\}_{\forall i,k}$, $\mathbf{p} \triangleq \{p_{i,k}\}_{\forall i,k}$, and $\boldsymbol{f} \triangleq \{f_{i,k}^{lo}\}_{\forall i,k}$ as the set constraints of computational offloading decision, transmit power, and CPU frequency. respectively. Therefore, the problem is formulated below.

$$\begin{aligned}(\mathrm{P1})\ \min_{\boldsymbol{\varphi},\boldsymbol{p},\boldsymbol{f}}\ & \sum_{k\in\mathcal{K}} \zeta_k \\ \text{s.t.}\ & C_1 : p_{i,k} \leq p_{\max}, \forall i,k \\ & C_2 : f_{i,k}^{lo} \leq f_{\max}^{lo}, \forall i,k \\ & C_3 : f_{i,k}^{es} \leq f_{\max}^{es}, \forall i,k \\ & C_4 : \sum_{k\in\mathcal{K}} \pi_{k,a} \leq \chi_{\max}, \forall i,k \\ & C_5 : \lambda_k^{tot} \leq \lambda_k^{\max}, \forall k \\ & C_6 : \xi_k \leq \xi_k^{\max}, \forall k \\ & C_7 : r_{i,k}^{dl,s} \geq r_{\min}, \forall i,k \\ & C_8 : \varphi_{i,k} \in (0,1), \forall i,k \\ & C_9 : 0 \leq i \leq t_k, 1 \leq k \leq 2.\end{aligned} \tag{25}$$

The constraints are described as follows: constraint C_1 denotes the upper limit of transmission power, denoted by $p_{\max}$. Constraint C_2 limits the maximum CPU frequency of the specified device to $f_{\max}^{lo}$. Constraint C_3 restricts the computational resources to $f_{\max}^{es}$. Constraint C_4 restricts the number of IoT devices that each ES can serve at $\chi_{\max}$. Constraint C_5 sets the maximum latency threshold at $\lambda_k^{\max}$. Constraint C_6 specifies the maximum energy threshold as $\xi_k^{\max}$. Constraint C_7 establishes the minimum rate requirement, represented as $r_{\min}$. Constraint C_8 characterizes the computational offloading decision. Constraint C_9 details the number of tasks executed on k^{th} device.

IV. Proposed Solution

The problem outlined is a MINLP combinatory optimization problem, which is computationally intractable. It is due to the combinatorial nature of binary variable $\boldsymbol{\varphi}$ and strong coupling

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

with other continuous optimization variables, e.g., transmit power and CPU frequency. This problem is challenging to solve directly using an exhaustive search, especially in a dense network. It is important to note that there is a direct relationship between CPU frequency and local computational time as indicated in (8). Similarly, there is a direct relationship between power and transmission time while offloading as indicated in (15). Therefore, optimizing the problem concerning the power and CPU frequency is equivalent to optimizing it over the time allocation variables, i.e., $\{\eta_{i,k}^{lo,s}\}$, $\{\eta_{i,k}^{es,s}\}$, respectively. This inequality implies that if we understand how the time allocation variables can be optimized, we can infer the optimal power and CPU frequency, simplifying the original problem. By introducing a temporary variable $\xi_t = \max\{\xi_{k_1}^{hold} + \xi_{k_2}^{hold}\}$, we equivalently transform (25) to

$$\begin{aligned}
(\text{P2}) \quad & \min_{\boldsymbol{\varphi},\{\eta_{i,k}^{lo,s}\},\{\eta_{i,k}^{es,s}\},\xi_t} \quad \sum_{k\in\mathcal{K}} \zeta_k \\
\text{s.t.} \quad & C_1 - C_9 \\
& C_{10}: \xi_t \geq \xi_{k_1}^{hold}, \xi_t \geq \xi_{k_2}^{hold}, \\
& C_{11}: \eta_{i,k}^{lo,s} \geq \tfrac{\delta_{i,k}}{f_{\max}^{lo}}, \\
& C_{12}: \eta_{i,k}^{es,s} \geq \tfrac{\beta_{i-1,k}}{r_{i,k}^{es,s}}, \\
& \text{where } p_{i,k}^{s} = p_{\max}.
\end{aligned} \tag{26}$$

Constraint C_{10} establishes an upper bound for ξ_t. This equation implies that if we have computed the optimal solution $\left\{\boldsymbol{\varphi}^*, \{\eta_{i,k}^{lo,s}\}^*, \{\eta_{i,k}^{es,s}\}^*\right\}$ of (26), then we can easily compute power and CPU frequency in (25). The problem (26) is non-convex due to the combinatorial nature of binary variable $\boldsymbol{\varphi}$. For given $\boldsymbol{\varphi}$, the remaining optimization over $\left(\{\eta_{i,k}^{lo,s}\}, \{\eta_{i,k}^{es,s}\}, \xi_t\right)$ becomes a convex problem.

A. *Optimal Transmit Power and local CPU Frequency for Given $\boldsymbol{\varphi}$*

We solve the above problem for the given offloading decisions to compute a closed-form solution of optimal transmit power and local CPU frequency. We approximate the objective function as an unbounded optimization problem by utilizing the concept of partial Lagrangian subject to constraint C_{10}.

$$\begin{aligned}
& L_p\left(\{\eta_{i,k}^{lo,s}\}, \{\eta_{i,k}^{es,s}\}, \xi_t, \mu_{k_1}, \mu_{k_2}\right) = \\
& \zeta_{k_1} + \zeta_{k_2} + \mu_{k_1}(\xi_{k_1}^{hold} - \xi_t) + \mu_{k_2}(\xi_{k_2}^{hold} - \xi_t),
\end{aligned} \tag{27}$$

where $\mu_{k_1} \geq 0$ and $\mu_{k_2} \geq 0$ are the Lagrange multipliers computed to satisfy all remaining constraints. If $\mu_{k_1}^{opt}$ and $\mu_{k_2}^{opt}$ represent the optimal values for Lagrangian's multiplier, we derive the analytical expressions to compute the optimal power and CPU frequencies for both IoT devices as in [29]. To analyze device k_1, the derivative of (27) with respect to $\eta_{i,k_1}^{lo,s}$ is computed as follows.

$$L_p^{'} = \frac{\partial L_p}{\partial \eta_{i,k_1}^{lo,s}}, \forall i, k, \tag{28}$$

$$\frac{\partial L_p}{\partial \eta_{i,k_1}^{lo,s}} = \partial_{k_1}^{\lambda} - \frac{2\mu\partial_{k_1}^{\xi}(\delta_{i,k_1})^3}{(\eta_{i,k_1}^{lo,s})^3} + \mu_{k_1}, \forall i, k, \tag{29}$$

where the above derivation indicates that it is a non-decreasing function for $\eta_{i,k_1}^{lo,s} \in \left[\frac{\delta_{i,k_1}}{f_{\max}^{lo}}, +\infty\right)$. Hence, if the first derivative is positive, then we have $f_{i,k_1}^{lo,opt} = f_{\max}^{lo}$. Otherwise,

$$\mu_{k_1} + \partial_{k_1}^{\lambda} = \frac{2\mu\partial_{k_1}^{\xi}(\delta_{i,k_1})^3}{(\eta_{i,k_1}^{lo,s})^3}, \forall i, \tag{30}$$

$$(\eta_{i,k_1}^{lo,s})^3 = \frac{2\mu\partial_{k_1}^{\xi}(\delta_{i,k_1})^3}{\mu_{k_1} + \partial_{k_1}^{\lambda}}, \forall i, \tag{31}$$

$$\eta_{i,k_1}^{lo,s} = \delta_{i,k_1}\left(\frac{2\mu\partial_{k_1}^{\xi}}{\mu_{k_1} + \partial_{k_1}^{\lambda}}\right)^{\frac{1}{3}}, \forall i. \tag{32}$$

$$f_{i,k_1}^{lo,opt} \Rightarrow \frac{\delta_{i,k_1}}{\eta_{i,k_1}^{lo,s}} = \left(\frac{\mu_{k_1}^{opt} + \partial_{k_1}^{\lambda}}{2\mu\partial_{k_1}^{\xi}}\right)^{\frac{1}{3}}, \forall i. \tag{33}$$

The optimal CPU frequencies are equivalent to the frequencies computed as follows. If $(\varphi_{i,k})_{\forall i,k} = 0$ and $i \in l$ then

$$f_{i,k_1}^{lo,opt} = \min\left(\sqrt[3]{\frac{\mu_{k_1}^{opt} + \partial_{k_1}^{\lambda}}{2\mu\partial_{k_1}^{\xi}}}, f_{\max}^{lo}\right), \tag{34}$$

for IoT device k_1. If $i \in \{1, 2, ..., |\text{t}|-1\}$ then

$$f_{i,k_2}^{lo,opt} = \min\left(\sqrt[3]{\frac{\mu_{k_2}^{opt}}{2\mu\partial_{k_2}^{\xi}}}, f_{\max}^{lo}\right), \tag{35}$$

for IoT device k_2. Otherwise, for $i \in \{|\text{t}|, ..., |\text{M}|\}$, we have

$$f_{i,k_2}^{lo,opt} = \min\left(\sqrt[3]{\frac{\partial_{k_2}^{\lambda}}{2\mu\partial_{k_2}^{\xi}}}, f_{\max}^{lo}\right). \tag{36}$$

Hence, we have the following observations:

- The optimal CPU frequencies for all similar tasks are identical. Specifically for device k_1, $i \in \{1, 2, ..., |\text{L}|\}$. For device k_2, the relevant tasks are either $i \in \{1, 2, ..., |\text{t}|-1\}$ or $i \in \{|\text{t}|, ..., |\text{M}|\}$, regardless of channel conditions.
- When the value of $\partial_{k_1}^{\lambda}$ or $\mu_{k_1}^{opt}$ is higher, the optimal strategy shifts towards accelerating local computations. Conversely, if $\partial_{k_1}^{\xi}$ is higher, the optimal strategy shifts involve conserving energy with a lower optimal CPU frequency.
- The value of $\mu_{k_2}^{opt}$ does not affect the optimal CPU frequency for tasks where $i \in \{|\text{t}|, ..., |\text{M}|\}$. Otherwise, the optimal CPU frequency increases proportionally with the value of the Lagrange multiplier.

The optimal power for IoT device k_1 is calculated by computing the derivative of the problem (27) with respect to $\eta_{i,k_1}^{es,s}$, when $i \leq |\text{M}|$, which is given by

$$\begin{aligned}
\frac{\partial L_p^{'}}{\partial \eta_{i,k_1}^{\text{es,s}}} &= \partial_{k_1}^{\lambda} + \\
\partial_{k_1}^{\xi}\left[\frac{1}{\|h_{i,k_1}^{s}\|^2} f\left(\frac{\beta_{i-1,k_1}}{\eta_{i,k_1}^{\text{es,s}}}\right) + \frac{\eta_{i,k_1}^{\text{es,s}}}{\|h_{i,k_1}^{s}\|^2} f'\left(\frac{\beta_{i-1,k_1}}{\eta_{i,k_1}^{\text{es,s}}}\right)\right] &+ \mu_{k_1},
\end{aligned} \tag{37}$$

$$\frac{\partial L_p^{'}}{\partial \eta_{i,k_1}^{es,s}} = \partial_{k_1}^{\lambda} + \partial_{k_1}^{\xi} \left[\frac{\bar{\sigma}^2}{\|h_{i,k_1}^s\|^2} 2^{\left(\frac{\beta_{i-1,k_1}}{w_s \eta_{i,k_1}^{es,s}}\right)} \left(1 - \frac{\beta_{i-1,k_1}}{w_s \eta_{i,k_1}^{es,s}} \ln 2\right) - \frac{\bar{\sigma}^2}{\|h_{i,k_1}^s\|^2} \right] + \mu_{k_1}. \tag{38}$$

It is worth mentioning that the above function is an increasing function. To check the curvature, the second order derivative for the problem (27) concerning $\eta_{i,k_1}^{es,s}$ is given by

$$\frac{\partial L_p^{''}}{\partial (\eta_{i,k_1}^{es,s})^2} = \partial_{k_1}^{\xi} \left[\frac{\bar{\sigma}^2}{\|h_{i,k_1}^s\|^2} \frac{(\beta_{i-1,k_1})^2}{(w_s)^2 (\eta_{i,k_1}^{es,s})^3} 2^{\left(\frac{\beta_{i-1,k_1}}{w_s \eta_{i,k_1}^{es,s}}\right)} (\ln 2)^2 \right] + \mu_{k_1} > 0, \tag{39}$$

which shows that the function is concave-up and non-decreasing for $\eta_{i,k_1}^{es,s} \in \left[\frac{\beta_{i-1,k_1}}{r_{i,k_1}^{es,s}}, +\infty\right)$, where where $p_{i,k_1}^s = p_{\max}$. Therefore, we have $\frac{\partial L_p^{'}}{\partial \eta_{i,k_1}^{es,s}} \in \left[\frac{\beta_{i-1,k_1}}{r_{i,k_1}^{es,s}}, \partial_{k_1}^{\lambda} + \mu_{k_1}\right]$ if $\frac{\partial L_p^{'}}{\partial \eta_{i,k_1}^{es,s}} > 0$, when $p_{i,k_1}^s = p_{\max}$. The optimal transmit powers are equivalent to the powers computed below. If $(\varphi_{i,k})_{\forall i,k} = 1$, the optimal power for device k_1 is given in (40). where $\omega(\psi_1)$ and $\omega(\psi_2)$ denote special inverse functions. These functions are defined as the inverse of the given function $y = f(x) = xe^x$, where x is expressed as $x = \omega(y)$. The computation of threshold values is performed as follows. For $i \in \{1, ..., |\mathrm{L}|\}$:

$$\upsilon_{i,k_1}^{threshold} = \frac{\bar{\sigma}^2}{p_{\max}} \left[\frac{\nu_1}{\omega(-\nu_1 e^{-\nu_1})} - 1 \right], \quad \text{where } \nu_1 = 1 + \left(\frac{\partial_{k_1}^{\lambda} + \mu_{k_1}^{opt}}{\partial_{k_1}^{\xi} p_{\max}} \right). \tag{41}$$

For $i = |\mathrm{L}|+1$:

$$\upsilon_{i,k_1}^{threshold} = \frac{\bar{\sigma}^2}{p_{\max}} \left[\frac{\nu_2}{\omega(-\nu_2 e^{-\nu_2})} - 1 \right], \quad \text{where } \nu_2 = 1 + \left(\frac{\mu_{k_1}^{opt}}{\partial_{k_1}^{\xi} p_{\max}} \right). \tag{42}$$

We compute the optimal power for the IoT device k_2 in (43). where $\omega(\psi_3)$ and $\omega(\psi_4)$ represent special functions, as defined above. We compute the threshold values as follows. For $i \in \{1, ..., |\mathrm{t}|\}$:

$$\upsilon_{i,k_2}^{threshold} = \frac{\bar{\sigma}^2}{p_{\max}} \left[\frac{\nu_3}{\omega(-\nu_3 e^{-\nu_3})} - 1 \right], \quad \text{where } \nu_3 = 1 + \left(\frac{\mu_{k_2}^{opt}}{\partial_{k_2}^{\xi} p_{\max}} \right). \tag{44}$$

For $i = \{|\mathrm{t}|+1, ..., |\mathrm{M}|\}$:

$$\upsilon_{i,k_2}^{threshold} = \frac{\bar{\sigma}^2}{p_{\max}} \left[\frac{\nu_4}{\omega(-\nu_4 e^{-\nu_4})} - 1 \right], \quad \text{where } \nu_4 = 1 + \left(\frac{\partial_{k_2}^{\lambda}}{\partial_{k_2}^{\xi} p_{\max}} \right). \tag{45}$$

Hence, we have the following observations:

- The channel gain $\|h_{i,k_1}^s\|^2$ is inversely proportional to the optimal transmit power when $\|h_{i,k_1}^s\|^2 > \upsilon_{i,k_1}^{\text{threshold}}$.
- The transmission power has been set to the maximum power, i.e., $p_{i,k_1}^{s,opt} = p_{\max}$ when $\|h_{i,k_1}^s\|^2 < \upsilon_{i,k_1}^{threshold}$.
- When the maximum transmit power $p_{\max}$ increases, it leads to a decrease in the squared magnitude of the channel gain h_{i,k_1}^s, indicating that the device is adapting to meet the conditions of a weaker channel.

B. *Bisection Method for Given $\boldsymbol{\varphi}$ to Obtain Optimal Power $\boldsymbol{p}$ and Frequency $\boldsymbol{f}$*

We can state that $\xi_{k_1}^{hold} \leq \xi_{k_2}^{hold}$ and $\mu_{k_2}^{opt}$ holds at the optimum of (26). It can be proved through contradiction. Assume that $\{\eta_{i,k}^{lo,s}, \tilde{\eta}_{i,k}^{es,s}\}$ are the optimal solutions with $\xi_{k_1}^{hold} > \xi_{k_2}^{hold}$. According to Karush Kuhn Tucker (KKT) conditions $\mu_{k_1^{opt}}(\xi_{k_1}^{hold} - \xi_t) = 0$, and $\mu_{k_2^{opt}}(\xi_{k_2}^{hold} - \xi_t) = 0$. Given that $\mu_{k_1}^{opt} > 0$ and $\mu_{k_2}^{opt} = 0$, it follows from Eq. (33) and (40) that the optimal CPU frequency $f_{i,k_1}^{lo,opt}$ and $p_{i,k_1}^{s,opt}$ are finite. This implies that $\{(\eta_{i,k_1}^{lo,s})^*, (\tilde{\eta}_{i,k_1}^{es,s})^*\}$ are also finite for each task, resulting in a finite $\xi_{k_1}^{hold}$. However, when $\mu_{k_2}^{opt} = 0$, we will have the optimal value $(\eta_{i,k_2}^{lo,s})^* \Rightarrow \infty$, i.e., for $i \in \{1, 2, 3, .., |t|-1\}$ from $f_{i,k_2}^{lo,opt}$. Similarly, it follows from Eq. (43) that optimal value $(\tilde{\eta}_{i,k_2}^{es,s})^* \Rightarrow \infty$, i.e., for $i \in \{1, 2, 3, .., |t|\}$, resulting in infinite $\xi_{k_1}^{hold}$. This contradicts our initial assumption that $\xi_{k_1}^{hold} > \xi_{k_2}^{hold}$. Hence, we have

$$\begin{cases} \text{if } i \in \{1, \ldots, |\mathrm{L}|\},\ p_{i,k_1}^{s,opt} = \begin{cases} p_{\max}, & \text{if } \|h_{i,k_1}^s\|^2 < \upsilon_{i,k_1}^{\text{threshold}}, \\ \frac{\bar{\sigma}^2}{\|h_{i,k_1}^s\|^2} \left[\frac{\psi_1}{\omega(\psi_1 e^{-1})} - 1 \right], & \text{otherwise}, \\ \text{where } \psi_1 = \left[\frac{\|h_{i,k_1}^s\|^2 (\partial_{k_1}^{\lambda} + \mu_{k_1}^{\text{opt}})}{\partial_{k_1}^{\xi} \bar{\sigma}^2} \right] - 1, \end{cases} \\ \text{if } i = |\mathrm{L}| + 1, \ldots, |\mathrm{M}|,\ p_{i,k_1}^{s,opt} = \begin{cases} p_{\max}, & \text{if } \|h_{i,k_1}^s\|^2 < \upsilon_{i,k_1}^{\text{threshold}}, \\ \frac{\bar{\sigma}^2}{\|h_{i,k_1}^s\|^2} \left[\frac{\psi_2}{\omega(\psi_2 e^{-1})} - 1 \right], & \text{otherwise}, \\ \text{where } \psi_2 = \left[\frac{\|h_{i,k_1}^s\|^2 \mu_{k_1}^{\text{opt}}}{\partial_{k_1}^{\xi} \bar{\sigma}^2} \right] - 1, \end{cases} \end{cases} \tag{40}$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

Algorithm 1 Bisection Method to Compute Optimal Power and Local Frequency

1: **Input:** $\boldsymbol{\varphi}$, precision factor $\epsilon = 0.001$.
2: **Output:** Optimal $\boldsymbol{f}$ and $\boldsymbol{p}$
3: Initialize $\wp^{'}_{ub} \leftarrow \partial^{\lambda}_{k_2}, \ \wp^{'}_{lb} \leftarrow 0$
4: **if** $\left(\xi^{hold}_{k_1} - \xi^{hold}_{k_2}\right)|_{\wp'=\wp^{'}_{lb}} < 0$ **then**
5:     Set $\wp' = \wp^{'}_{lb}, \mu_{k_1} = \wp', \mu_{k_2} = \partial^{\lambda}_{k_2} - \wp'$
6:     Compute $\boldsymbol{f}$ using (33)-(36) and $\boldsymbol{p}$ using (40)-(43)
7: **else**
8:     **while** $\left(\xi^{hold}_{k_1} - \xi^{hold}_{k_2}\right) < \epsilon$ **do**
9:         $\wp' = \left(\wp^{'}_{ub} + \wp^{'}_{lb}\right)/2$
10:         Set $\mu_{k_1} = \wp', \mu_{k_2} = \partial^{\lambda}_{k_2} - \wp'$
11:         Compute $\boldsymbol{f}$ using (33)-(36) and $\boldsymbol{p}$ using (40)-(43)
12:         **if** $\left(\xi^{hold}_{k_1} - \xi^{hold}_{k_2}\right) < 0$ **then**
13:             Set $\wp^{'}_{ub} \leftarrow \wp'$
14:         **else**
15:             Set $\wp^{'}_{lb} \leftarrow \wp'$
16:         **end if**
17:     **end while**
18: **end if**

$\xi^{hold}_{k_1} \leq \xi^{hold}_{k_2}$, and $\max(\xi^{hold}_{k_1}, \xi^{hold}_{k_2}) = \xi^{hold}_{k_2}$. Therefore, the optimization problem (28) is simplified as below.

$$\begin{aligned} \text{(P3)} \ \min_{\boldsymbol{\varphi},\boldsymbol{p},\boldsymbol{f}} \quad & \sum_{k \in \mathcal{K}} \zeta_k \\ \text{s.t.} \quad & C_1 - C_9 \\ & C_{13} : \xi^{hold}_{k_1} \leq \xi^{hold}_{k_2}. \end{aligned} \tag{46}$$

Similarly, the Lagrangian for the problem defined in Eq. (46) subject to constraint C_{13} is given by

$$L^{p}_{2}\left(\boldsymbol{p}, \boldsymbol{f}, \wp'\right) = \zeta_{k_1} + \zeta_{k_2} + \wp'(\xi^{hold}_{k_1} - \xi^{hold}_{k_2}), \tag{47}$$

whereas $\wp' \geq 0$ represents the Lagrange multiplier that satisfies all remaining constraints. Various iterative solutions can be applied to solve Eq. (47), but we apply KKT conditions [29]. The details are omitted here. Algorithm (1) presents the bisection search method to compute optimal power and frequency. This algorithm accepts the matrix $\boldsymbol{\varphi}$ and a precision factor ϵ as inputs and yields the optimal values of $\boldsymbol{f}$ and $\boldsymbol{p}$ as outputs. In line 3, the upper and lower bounds are initialized. Then, the inequality constrained C_{13} is checked, and the values for Lagrange multipliers are updated in line 5. Initial values for frequency and power are computed in line 6. The loop then iterates until the condition $\left(\xi^{hold}_{k_1} - \xi^{hold}_{k_2}\right) < \epsilon$ is satisfied, as outlined in lines 8-17. The $\wp^{'}$ value is updated in line 9 during this iteration. Line 11 involves computing the optimal frequency and power. Importantly, from lines 12 to 16, if the difference $\xi^{hold}_{k_1}$ and $\xi^{hold}_{k_2} < 0$ results in a negative value, the algorithm updates the upper bound of the variable, designated as $\wp^{'}_{ub}$. Conversely, the algorithm converges towards the lower bound variable, $\wp^{'}_{lb}$. The overall complexity of the Algorithm (1) is $\mathcal{O}(L + M)$.

Algorithm 2 BSSE Algorithm for Optimal offloading

Input: Maximum iterations $(i_{\max})$, Number of task of each device $(t_k)_{\forall k \in \{k_1,k_2\}}$.
Output: Matrix $\boldsymbol{\varphi}$
Set $t \leftarrow 0, \ \zeta_{k_1} \leftarrow \infty$, and $\zeta_{k_2} \leftarrow \infty$
We initialize $\boldsymbol{\varphi}$ randomly either with the value 0 or 1
while Convergence or $t < i_{\max}$ **do**
    Set $i \leftarrow 0$
    while $i < t_{k_1}$ **do**
        Set flag $\leftarrow$ True
        j $\leftarrow 0$
        while $j < t_{k_2}$ **do**
            if flag == False **then**
                break
            end if
            $\varphi^{'} \leftarrow \boldsymbol{\varphi}$
            Update $(\boldsymbol{\varphi}, i, j) \leftarrow$ Swap $\boldsymbol{\varphi}[i, k_1]$ and $\boldsymbol{\varphi}[j, k_2]$
            Compute $\boldsymbol{f}$ and $\boldsymbol{p}$ using Algorithm (1)
            if $\left(\zeta^{\text{new}}_{k_1} + \zeta^{\text{new}}_{k_2}\right) \leq (\zeta_{k_1} + \zeta_{k_2})$ **then**
                Set $\zeta_{k_1} \leftarrow \zeta^{\text{new}}_{k_1}$
                Set $\zeta_{k_2} \leftarrow \zeta^{\text{new}}_{k_2}$
                flag $\leftarrow$ False
            else
                $\boldsymbol{\varphi} \leftarrow \varphi^{'}$
            end if
            Set $j \leftarrow j + 1$
        end while
        Set $i \leftarrow i + 1$
    end while
    Set $t \leftarrow t + 1$
end while

C. *Optimizing Offloading Decision $\boldsymbol{\varphi}$ using Given $\boldsymbol{p}$ and $\boldsymbol{f}$*

The optimal offloading scheme adheres to the one climb policy [29], implying that there will be at most one instance

$$\begin{cases} \text{if } i \in \{1, \ldots, |\mathrm{t}|\}, \ p^{s,opt}_{i,k_2} = \begin{cases} p_{\max}, & \text{if } \|h^{s}_{i,k_2}\|^2 < \upsilon^{\text{threshold}}_{i,k_2}, \\ p^{s,opt}_{i,k_2} = \frac{\bar{\sigma}^2}{\|h^{s}_{i,k_2}\|^2}\left[\frac{\psi_3}{\omega(\psi_3 e^{-1})} - 1\right], & \text{otherwise}, \\ \text{where } \psi_3 = \left[\frac{\|h^{s}_{i,k_2}\|^2 \mu^{\text{opt}}_{k_2}}{\partial^{\xi}_{k_2}\bar{\sigma}^2}\right] - 1, \end{cases} \\ \text{if } i = \{|\mathrm{t}| + 1, \ldots, |\mathrm{M}|\}, \ p^{s,opt}_{i,k_2} = \begin{cases} p_{\max}, & \text{if } \|h^{s}_{i,k_2}\|^2 < \upsilon^{\text{threshold}}_{i,k_2}, \\ p^{s,opt}_{i,k_2} = \frac{\bar{\sigma}^2}{\|h^{s}_{i,k_2}\|^2}\left[\frac{\psi_4}{\omega(\psi_4 e^{-1})} - 1\right], & \text{otherwise}, \\ \text{where } \psi_4 = \left[\frac{\|h^{s}_{i,k_2}\|^2 \partial^{\lambda}_{k_2}}{\partial^{\xi}_{k_2}\bar{\sigma}^2}\right] - 1, \end{cases} \end{cases} \tag{43}$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

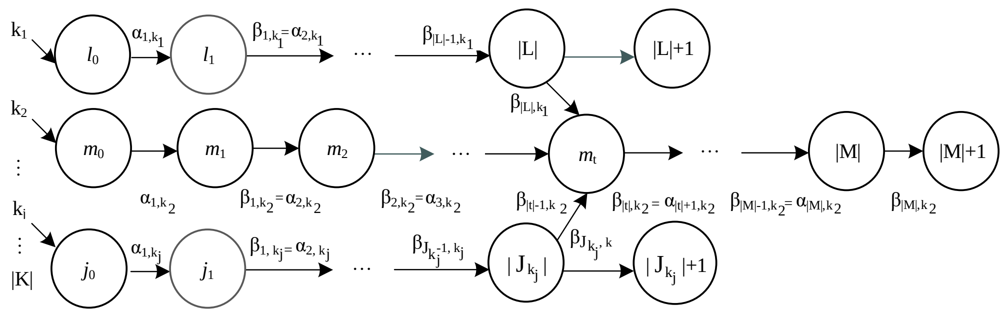

Fig. 2. Task dependent computational model incorporating outputs from multiple devices.

where $\varphi_{i,k} = 1$ and $\varphi_{i-1,k} = 0$. An approximate optimal offloading solution utilizing local and bisection search techniques is proposed in Algorithm (2). Initially, we set the maximum iteration count to zero, and the energy-time cost for both IoT devices to infinity denoted as $\zeta_{k_1} = \zeta_{k_2} = \infty$ in line 3. Afterward, the proposed algorithm initiates with a random offloading scheme and progressively refines it towards minimizing the overall energy-time cost, as detailed in lines 5 to 29. The process includes swapping $\boldsymbol{\varphi}[i, k_1]$ and $\boldsymbol{\varphi}[j, k_2]$ within the matrix $\boldsymbol{\varphi}$ and updating the tuple $(\boldsymbol{\varphi}, i, j)$ in line 14 and 15. Line 16 invokes the bisection algorithm, which computes optimal frequency and power using equations (33), (36), (40), and (43), respectively. The outcomes of the bisection search are then evaluated and compared with previously optimized energy-time cost values, as shown in lines 17-23. If $\left(\zeta_{k_1}^{\text{new}} + \zeta_{k_2}^{\text{new}}\right) \leq \left(\zeta_{k_1} + \zeta_{k_2}\right)$ is true, then $\zeta_{k_1} \leftarrow \zeta_{k_1}^{\text{new}}$ and $\zeta_{k_2} \leftarrow \zeta_{k_2}^{\text{new}}$; otherwise, the matrix $\boldsymbol{\varphi}$ reverts to $\varphi^{'}$. This process iterates until the solution convergences at optimized values.

D. Complexity Analysis

The first benchmark is the naive search algorithm corresponding to the complexity of $\mathcal{O}\left(16^{\text{L+M}}\right)$, as it considers each possible case for both devices with L and M tasks, i.e., 2^4 binary decisions. It algorithm becomes increasingly inefficient with a more significant number of tasks, especially when $|\text{L}| \approx 10$ and $|\text{M}| \approx 10$. To mitigate this, we effectively narrow the search space for both devices, i.e., $\left(\frac{(L+1)L}{2} + 1\right)\left(\frac{(M+1)M}{2} + 1\right)$. This includes the scenario where the device is unable to offload data during the entire duration of the execution. This approach yields an optimal offloading scheme with a complexity of $\mathcal{O}\left(\text{L}^2\text{M}^2\right)$. However, its efficiency diminishes for large values, precisely when $|\text{L}| > 100$ and $|\text{M}| > 100$. Upon examining the pseudo-code provided, it becomes evident that the overall complexity of the BSSE algorithm is $\mathcal{O}\left(\text{LM}\right)$, significantly lesser than the benchmarks. Therefore, the proposed algorithmic solution is regarded as near-optimal, approximating the performance of the optimal offloading scheme.

V. Multi-Device Scenario

In Fig. 2, we extend the proposed model to support multiple devices. In this scenario, the inputs for an intermediate task t on IoT device k_2 require the final task outputs from all other $|\text{K}|-1$ devices. For example, for IoT device k_2, we define $\alpha_{t,k_2} = \beta_{|t|-1,k_2} + \sum\limits_{k\in\mathcal{K}, k_j \neq k_2} \beta_{J_{k_j}, k_j}$, where J_{k_j} denotes the sequential number of tasks needs to execute on device k_j. The first terms on the right side of the equation represent the output from the previous task $|t|-1$. Meanwhile, the summation term adds up the outputs of the final tasks from all other devices (excluding k_2), which are required for the intermediate task t. The waiting time for the output of the J_{k_j}-th task to reach IoT device k_2 is calculated in (48). The waiting time needed before the joint task can be executed is determined by the total waiting time $\xi^{hold} = \max\left\{\xi_{k_1}^{hold}, \xi_{k_2}^{hold}, ..., \xi_{|k_j|}^{hold}, ..., \xi_{|\text{K}|}^{hold}\right\}$. Therefore, the problem (26) is reformulated as below.

$$
\begin{aligned}
&\text{(P4)} \quad \min_{\boldsymbol{\varphi}, \{\eta_{i,k_j}^{\text{lo,s}}\}, \{\eta_{i,k_j}^{\text{es,s}}\}, \xi_t} \sum_{k\in\mathcal{K}} \zeta_k \\
&\text{subject to} \\
&\quad C_1 - C_9 \\
&\quad C_{10} : \xi_t \geq \xi_{k_1}^{\text{hold}}, \xi_t \geq \xi_{k_2}^{\text{hold}}, \ldots, \xi_t \geq \xi_{k_j}^{\text{hold}}, \ldots, \xi_t \geq \xi_{|\text{K}|}^{\text{hold}}, \\
&\quad C_{11} : \eta_{i,k_j}^{\text{lo,s}} \geq \frac{\delta_{i,k_j}}{f_{i,k_j}^{\text{lo}} - \hat{f}_{i,k_j}^{\text{lo}}}, \\
&\quad \text{where } f_{i,k_j}^{\text{lo}} = f_{i,k_j}^{\text{lo,max}}, \\
&\quad C_{12} : \eta_{i,k_j}^{\text{es,s}} \geq \frac{\beta_{i-1,k_j}}{r_{i,k_j}^{\text{es,s}}}, \\
&\quad \text{where } p_{i,k_j}^{\text{s}} = p_{\max}.
\end{aligned} \tag{49}
$$

We assume that the ES is equipped with c_r processor cores, where each core is exclusively assigned to the execution of an individual task operating at a constant frequency of $f_{\max}^{es}$. Therefore, if $J_{\max}$ represents the upper limit on the number of tasks that can be processed simultaneously on the ES, the total count of cores must be adequate to meet this requirement, e.g., $J_{\max} \leq c_r$.

VI. Simulation and Results

A. Simulations Parameters

Regarding communication considerations, the IoT devices are uniformly distributed within the coverage area of the serving ABS, encompassing a radius ranging from 10 to 30 meters.

TABLE I
SIMULATION PARAMETERS.

Parameters	Values
Effective switched capacitance μ [29]	10^{-26}
Bandwidth W [29]	2 MHz
Intermediate task t [29]	4
Peak transmit power of each IoT device $p_{\max}$	200 mW
Tasks for device (k_1 and k_2) (L, M) [29]	$(3, 5)$
Power of ABS $(p_a)_{a\in\mathcal{A}}$ [36]	1 W
Noise power σ^2 [29]	10^{-10} W
ES speed $f_{\max}^{es}$ [29]	10^{10} Cycles/s
Error probability $\epsilon_{i,k}$ [36]	10^{-3}
Maximum delay requirement $\lambda_k^{\max}$ [16]	2 Seconds
Maximum EC requirement $\xi_k^{\max}$ [16]	1 Joule
Minimum data rate requirement $r_{\min}$ [16]	2 Bits/s
Peak computational frequency of device $f_{\max}^{lo}$ [16]	10^8 Cycles/s
Maximum IoT devices served by each ES $\chi_{\max}$ [16]	6
Distances of the respective devices(d_{k_1}, d_{k_2})	(15, 15) Meters
Blocklength value $(b_{i,k})_{k\in\{k_1,K_2\}}$ [36]	100
Deviation between real value of ES speed $\hat{f}_{i,k}^{es}$ [29]	2 %
Computing workload for device k_1 $\{\delta_{i,k_1}\}$ [29]	[65.5,40.3,96.6] (Mcycles)
Output data size for device k_1 $\{\alpha_{i,k_1}\}$ [29]	[1500,1000,1600, 1000, 0] Kbytes
Computing workload for device k_2 $\{\delta_{i,k_2}\}$ [29]	[70.8,95.3,86.4, 18.6,158.6] (Mcycles)
Output data size for device k_2 $\{\alpha_{i,k_2}\}$ [29]	[2000,1500,1000, [1400,1000],1500, 1000, 0] Kbytes

The computing requirements for the corresponding IoT devices are represented by the values $\{\delta_{i,k_1}\} = [65.5, 40.3, 96.6]$ (Mcycles) and $\{\delta_{i,k_2}\} = [70.8, 95.3, 86.4, 18.6, 158.6]$ (Mcycles). The total power budget for the MEC-enabled ABS under consideration is set at $(p_a)_{a\in\mathcal{A}} = 1$ W. Each IoT device's maximum transmit power is capped at $p_{\max} = 200$ mW. It is assumed that the input of the fourth task at IoT device k_2 depends upon the final task outputs from the other IoT devices. Assuming $f_{i,k}^{es} > f_{\max}^{lo}$, the peak computational frequency for each IoT device and the processing speed of each ES are established at $f_{\max}^{lo} = 10^8$ Cycles/s and $f_{\max}^{es} = 10^{10}$ Cycles/s, respectively. The FBL is set at 100, and the computing efficiency parameter for each device is defined as $\mu = 10^{-26}$ [29].

The system bandwidth is set at $W = 2$ MHz, while the noise spectral density is specified as $\sigma^2 = 10^{-10}$ Watts. The proportion of the anticipated processing rate is a pre-configured value, and its selection is based on established computing models and standard practices in wireless networking [15]. For the URLLC, the decoding error probability is defined as $\epsilon_{i,k} = 10^{-3}$. The maximum EC is limited at $\xi_k^{\max} = 0.5$ Joule. The experimental scenario is simulated using single ABS $\mathcal{A} = 1$ along with multiple IoT devices $\mathcal{K} = \{1, 2\}$ unless explicitly stated otherwise. The analytical assessment considers a maximum latency of 1 millisecond. The minimum time required to transmit a unit blocklength is 0.01 milliseconds, as indicated in reference [36]. The altitude of the serving ABs is fixed at 50 Meters. The pathloss models employed are detailed in [15]. The details of the simulation parameters are provided in Table I.

B. Performance Comparison

In Fig. 3, we study the correlation between devices. Specifically, we analyze the trade-off between total execution time and energy for each IoT device by varying the $\partial_{k_2}^{\lambda}$. For given $\partial_{k_1}^{\lambda}$, an increase in $\partial_{k_2}^{\lambda}$ results in increased EC while simultaneously resulting reducing the total execution time. This trade-off is also observable for IoT device k_1. The trade-off curve for IoT device k_1 converges to a critical point at $\partial_{k_1}^{\lambda} = 0.4$. This convergence indicates that beyond this value, the optimal execution time and energy of IoT device k_1 remains constant despite increases in $\partial_{k_2}^{\lambda}$. The underlying rationale behind this phenomenon can be attributed to the fact that with an increase in $\partial_{k_1}^{\lambda}$, the device k_1 operates in a dual capacity, i.e., assisting IoT device k_2 while concurrently minimizing its processing delay.

In Fig. 4, we illustrate the correlation between the devices and analyze the proposed scenario in depth. Specifically, the effect of the intermediate task t on the computational delay is explored. We observe that the waiting time for device k_1 increases with an increase in t when $\partial_{k_1}^{\lambda}$ is kept small, i.e., $\partial_{k_1}^{\lambda} = 0.05$ or 0.3. In contrast, the waiting time for device k_2 decreases. This is because device k_1 needs to complete all three tasks to meet the finish time for the first t tasks of device k_2, especially for smaller values of t, i.e., when $t = 1$ or 2. Meanwhile, device k_2 only needs to slow down its computations to get the final output from device k_1. Generally, this results in larger $\lambda_{k_1}^{tot}$ and smaller $\lambda_{k_2}^{tot}$ with increasing t. When $\partial_{k_1}^{\lambda}$ is further increased, i.e., $\partial_{k_1}^{\lambda} = 0.5$, computations for both devices become independent and are optimized separately, instead of minimizing its computational delay to meet the stringent requirements of t task at device k_2. Therefore, there is no change in the computational delay for both devices when t increases from 1 to 5.

$$\xi_{|k_j|}^{hold} = \overbrace{\sum_{j=1}^{J_{k_j}} \Big[(1-\varphi_{j,k_j})\eta_{j,k_j}^{lo,s} + \varphi_{j,k_j}(\tilde{\eta}_{j,k_j}^{es,s} + \eta_{j,k_j}^{es,s}) + \varphi_{j-1,k_j}\eta_{j,k_j}^{dl,s} - \varphi_{j-1,k_j}\varphi_{j,k_j}(\eta_{j,k_j}^{dl,s} + \eta_{j,k_j}^{es,s}) \Big]}^{\text{Total amount of time for } J_{k_j} \text{ task}}$$
$$+ \overbrace{\Big(1-\varphi_{J_{k_j},k_j}\Big)\eta_{J_{k_j}+1,k_j}^{es,s} + (1-\varphi_{t,k_2})\frac{\beta_{J_{k_j},k_j}}{r_{t,k_2}^{dl,s}}}^{\text{Transmission time of the output of } |\mathrm{J}| \text{ tasks by IoT device } k_j} . \quad (48)$$

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

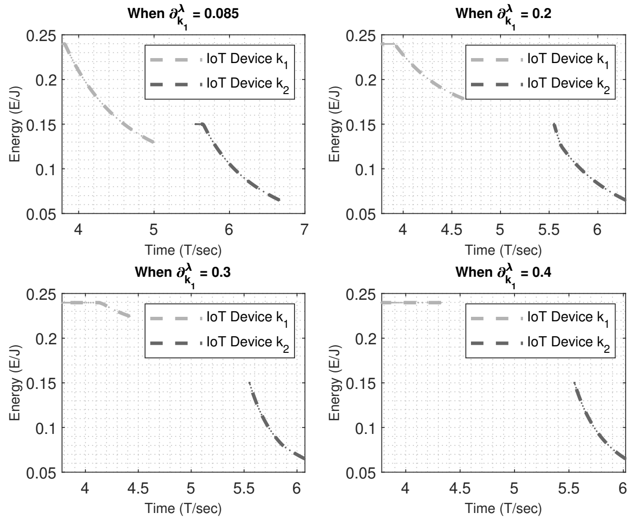

Fig. 3. The trade-off between the total execution time and energy for each IoT device when $\partial^{\lambda}_{k_2}$ varies, where $d_{k_1} = d_{k_2} = 15$ meters.

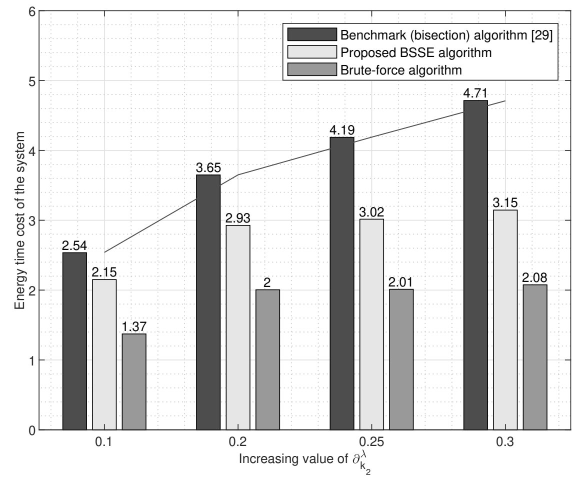

Fig. 5. Energy-time cost of benchmark and proposed algorithms, when $\mathcal{A} = 2$, $\mathcal{K} = 2$ and $d_{k_1} = d_{k_2} = 15$ meters.

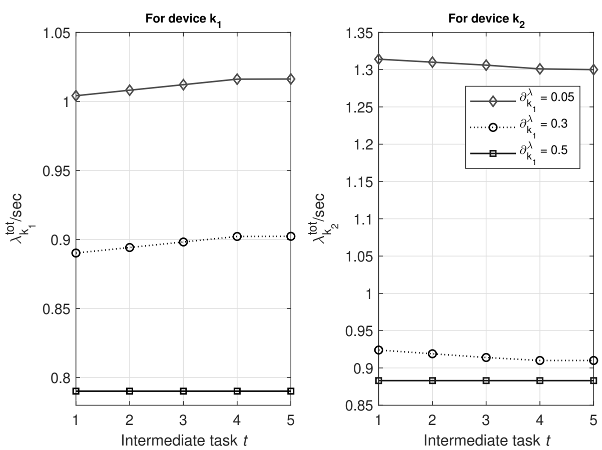

Fig. 4. Impact of Intermediate task t on the computational delay, when $(L, M, t) = (3, 5, t)$.

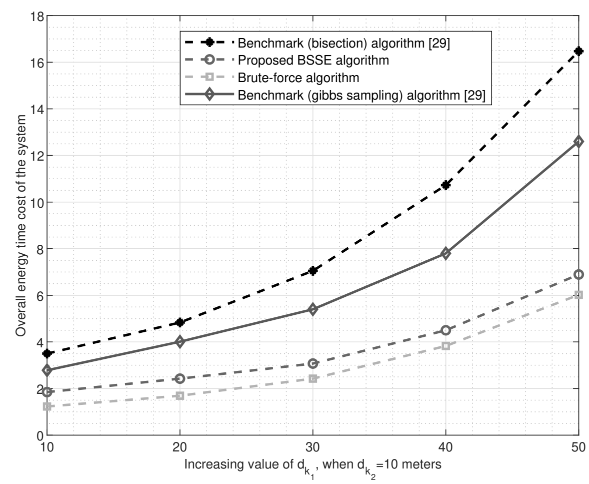

Fig. 6. Overall energy-time cost versus value of d_1, when $d_2 = 10$ meters.

Recall that in IoT device k, the weight are related by the formula $\partial^{\xi}_{k} = 1 - \partial^{\lambda}_{k}$. The system's energy-time cost objective value, varying with $\partial^{\lambda}_{k_2}$ is analyzed in Fig. 5. We employ two benchmarks against our proposed scheme to ensure a fair comparison. Our proposed BSSE algorithm is a modified version of the bisection algorithm, as referenced in [29]. It follows the same steps as defined in [29], except for line 3 in Algorithm (2), which is used to set the offloading factor to a random offloading scheme, and then update it to minimize the total energy-time cost objective. As anticipated, an increase in $\partial^{\lambda}_{k_2}$ leads to a higher energy-time cost of the system. Compared to the baseline algorithm in [29], our proposed BSSE algorithm is more efficient in terms of the energy-time cost of the system. The solutions obtained from the benchmark algorithm and those generated by the proposed BSSE algorithm have a significant difference. Furthermore, both the benchmark and the proposed algorithms satisfy the one-climb policy, i.e., (0 followed by 1) can appear only once. This finding underscores the potential of the proposed BSSE algorithm as a highly advantageous alternative to the benchmark algorithms, particularly in identifying the optimal offloading scheme in the counterparts.

In Fig. 6, we illustrate the impact of the increasing value of d_{k_1} on the overall energy-time cost of the system, where the distance of IoT device k_2 is fixed at $d_{k_2} = 10$ meters. The values of $\{\delta_{i,k_1}\}$ and $\{\delta_{i,k_2}\}$ are uniformly distributed between 10 and 200 Mcycles. The average performance of twenty independent iterations has been computed to plot this figure. It can be seen that the overall energy-time cost value increases with the distance of IoT device k_2 for all four schemes. As mentioned in Section II-I, this increase is attributed to the system's reliance on local computing. In this case, the IoT device k_1 needs to offload the output of the final task to the ES, which then forwards this information to IoT device k_2, resulting in a higher energy-time cost. Numerically, the energy-time cost of the proposed BSSE is slightly higher than the brute-force

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. 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/TGCN.2024.3425442

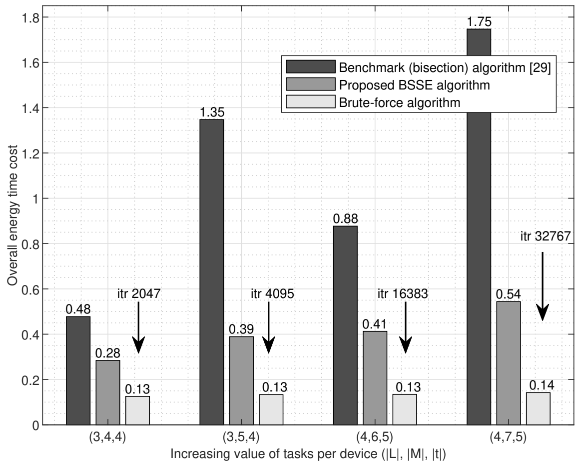

Fig. 7. Energy-time cost for increasing value of $(|L|, |M|, |t|)$ in three algorithms.

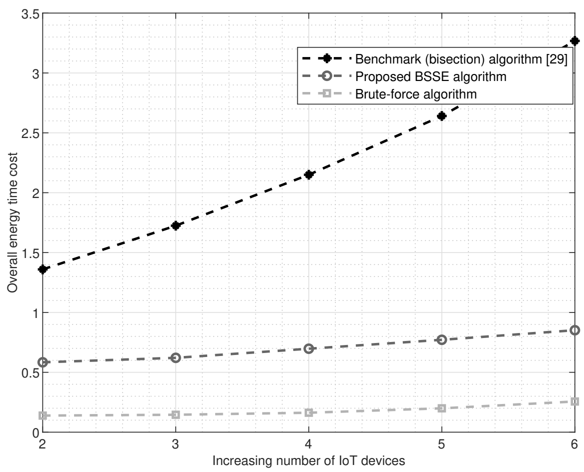

Fig. 8. Proposed task dependency model for a multi-device case study with $k = 4$.

algorithm, though the difference is not substantial. Moreover, the effectiveness of the proposed BSSE algorithm surpasses that of both bisection algorithm [29] and Gibbs sampling algorithm [29], i.e., around 89.39% to 138.96% lower and around 33.61% to 45.28% lower, respectively. This highlights the benefits of employing a strategy that optimizes resource allocation and offloading decisions for both IoT devices.

In Fig. 7, we further study the impact of different case studies (task dependency models) on system performance, considering various topological call graphs $(|L|, |M|, |t|)$. Each IoT device is constrained to a distance of 10 meters, where $p_{k_1} = p_{k_2} = 0.2$ W, $\partial^{\lambda}_{k_1} = 0$, and $\partial^{\lambda}_{k_2} = 0.5$. We initially examined scenarios involving three and four tasks on each device, with task four on device k_2 requiring the final output from device k_1. Subsequently, we varied the number of tasks and the position of the intermediate node to observe how this inter-device dependency influences the overall energy-time cost of the network at the optimal configuration. It is observed that the proposed BSSE algorithm achieves lower computational cost than the bisection algorithm and performs comparably to the brute-force algorithm. The effectiveness of the proposed algorithm can also be seen from its stable increase in energy-time cost value. Specifically, when we extend the call graph, the brute-force algorithm shows an exponential rise in the complexity growth. In contrast, the bisection algorithm solves it in polynomial time but with a higher energy-time cost of the system. However, the BSSE algorithm maintains lower fluctuations and stable increases. For instance, with the call graph $(|\mathrm{L}|, |\mathrm{M}|, |\mathrm{t}|) = (3, 5, 4)$, the optimal solution computed by bisection algorithm involves 4095 calls, yielding $\varphi_{l,k_1} = \{0\ 1\ 1\}$ and $\varphi_{m,k_2} = \{1\ 1\ 0\ 1\ 1\}$ with an energy-time cost of $\zeta_{k_1} + \zeta_{k_2} = 0.13$. In contrast, the proposed BSSE algorithm requires only five bisection algorithm calls to find a near-optimal solution, with $\varphi_{l,k_1} = \{1\ 0\ 1\}$ and $\varphi_{m,k_2} = \{0\ 0\ 1\ 1\ 1\}$, resulting in an energy-time cost of $\zeta_{k_1} + \zeta_{k_2} = 0.28$. Hence, the proposed algorithm is not only less computationally intensive but also achieves a 41.67% lower energy-time cost objective value compared to the benchmark bisection algorithm [29].

In Fig. 8, we extend the analysis of the overall energy-time cost of the proposed task dependency model to a multi-user case for all three schemes. The IoT devices are uniformly distributed at distances ranging from 10 to 30 meters. We assume the following computational requirements $\{\delta_{i,k_3}\} = [50.5, 45.3, 86.6]$ (Mcycles) and $\{\alpha_{i,k_3}\} = [1400, 1200, 1500, 1300]$ (Mcycles) for IoT device 3, $\{\delta_{i,k_4}\} = [65.5, 50.3, 75.6]$ (Mcycles) and $\{\alpha_{i,k_4}\} = [1500, 1400, 1000, 1500]$ (Mcycles) for IoT device 4, $\{\delta_{i,k_5}\} = [55.5, 42.3, 90.6]$ (Mcycles) and $\{\alpha_{i,k_5}\} = [1600, 1500, 1300, 1700]$ (Mcycles) for IoT device 5, and $\{\delta_{i,k_6}\} = [58.5, 47.3, 82.6]$ (Mcycles) and $\{\alpha_{i,k_6}\} = [1200, 1300, 1600, 1600]$ (Mcycles) for IoT device 6. The input required for the fourth task at IoT device k_2 requires the final task output from all other $|K|-1$ devices. It is worth noting that the BSSE algorithm exhibits superior performance compared to the baseline algorithm, as referenced in [29]. More precisely, the total energy-time cost metric of the system exhibits a rise in the curve, increasing from 0.7758 to 2.4160 when we increase the number of IoT devices from 2 to 6. This observed pattern depicts the intensified interdependence of tasks among the devices, yielding more substantial system performance improvements.

VII. Conclusion and Future Work

This study examines how task dependencies among IoT devices affect task offloading and resource allocation decisions. It does so by digitizing real-time edge networks and integrating aerial terrestrial networks. The presented problem is characterized as a mixed-integer non-linear programming problem. It becomes computationally intractable due to its inherently combinatorial linkage with task-offloading decisions and strong correlation with resource allocation. We propose a joint optimization approach that optimizes transmit power, CPU frequency, and task offloading policy to minimize the energy-time cost. Our proposed scheme can accommodate various tasks, sometimes exceeding one hundred. We observe that the system's energy-time cost closely approximates that of the brute-force algorithm, which delivers the optimal solution. A notable discovery is that our proposed BSSE algorithm demonstrates approximately equal energy-time cost for both devices, i.e., $\zeta_{k_1} \sim \zeta_{k_2}$, then brute-force algorithm where $\zeta_{k_1} > \zeta_{k_2}$. This similarity is advantageous as it ensures equitable energy-time costs for IoT devices, irrespective of their task loads, leading to significant cost savings for devices handling more tasks. Furthermore, the BSSE algorithm achieves convergence in just five iterations using the bisection method, a stark improvement over the brute-force algorithm's requirement of 4096 iterations. Our simulation results corroborate the effectiveness of the proposed algorithm compared to benchmark approaches. Future research will explore the potential of artificial intelligence in devising optimal offloading decisions that can swiftly adapt to changing channel conditions.

References

[1] X. Chi, H. Yu, P. Zeng, and Y. Li, "Towards critical industrial wireless control: Prototype implementation and experimental evaluation on URLLC," *IEEE Commun. Mag.*, pp. 1–7, 2023.

[2] S. Sundaresan, S.Maruthu, L. S. Kumar, and D. N. K. Jayakody, "Outage Analysis of Sparse Vector Coding based Downlink Multicarrier NOMA for URLLC," *IEEE Internet of Things J.*, pp. 1–1, 2023.

[3] X. Yuncong, P. Ren, and D. Xu, "Security-Oriented Pilot and Data Transmission for URLLC in Mission-Critical IoT Scenarios," *IEEE Internet of Things J.*, pp. 1–1, 2023.

[4] S. M. Muhammad, M. A. Tariq, J. Seo, and D. Kim, "An Overview of 3GPP Release 17 & 18 Advancements in the Context of V2X Technology," *Int. Conf. on Artif. Intell. in Informtion and Commun.*, pp. 057-062, 2023.

[5] Z. Hosein, N. Gholipoor, M. R. Mili, M. Rasti, H. Tabassum, and E. Hossain, "Resource Management for Multiplexing eMBB and URLLC Services Over RIS-Aided THz Communication," *IEEE Trans. on Commun.*, 71(2), pp. 1207–1225, 2023.

[6] L. Xiaocui, Z. Zhou, Q. He, Z. Shi, W. Gaaloul, and S. Yangui. "Re-Scheduling IoT Services in Edge Networks," *IEEE Trans. on Net. and Service Management*, pp 1-1, 2023.

[7] M. Weihao, K. Xiong, Y. Lu, P. Fan, and Z. Ding, "Energy Consumption Minimization in Secure Multi-antenna UAV-assisted MEC Networks with Channel Uncertainty," *IEEE Trans. on Wireless Commun.*, pp 1-1, 2023.

[8] L. Baozhu, F. Hou, G. Yang, H. Zhao, and S. Chen, "Data Analysis-Oriented Stochastic Scheduling for Cost-Efficient Resource Allocation in NFV Based MEC Network," *IEEE Trans. on Veh. Tech.*, pp 1-14, 2023.

[9] C. Yao, Y. Zhang, H. Luo, T. Chen, G. Chen, H. Chen, Y. Wang, H. Wei, and C. Chou, "Management and Orchestration of Edge Computing for IoT: A Comprehensive Survey," *IEEE Internet of Things J.*, pp 1-1, 2023.

[10] A. A. Nasir, H. D. Tuan, H. 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.

[11] L. Qi, X. Xu, X. Wu, Q. Ni, Y. Yuan, and X. Zhang. "Digital-Twin-Enabled 6G Mobile Network Video Streaming Using Mobile Crowdsourcing," *IEEE J. on Sel. Area. in Commun. (JSAC)*, Vol. 41, No. 10, pp. 3161 - 3174, Oct 2023.

[12] Z Yining, H. Zhang, X. Liu, C. Zhao, and F. Xu, "Digital twin-enabled deep reinforcement learning for joint scheduling of ultra-reliable low latency communication and enhanced mobile broadband: A reliability-guaranteed approach," *Trans. on Emerging Telecommu. Tech.* 34(3), 2023.

[13] W. Yuntao, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu., "A Survey on Digital Twins: Architecture, Enabling Technologies, Security and Privacy, and Future Prospects," *IEEE Internet of Things J.*, pp 1-1, 2023.

[14] T. Liu, L. Tang, W. Wang, Q. Chen, X. Zeng, "Digital twin assisted task offloading based on edge collaboration in the Digital Twin Edge Network, " *IEEE Internet of Things J.*, 9(2), pp. 1427–1444.

[15] T. Q. Duong, D. V. Huynh, Y. Li, E. G. Palacios, K. Sun, "Digital Twin-enabled 6G aerial edge computing with ultra-reliable and low-latency communications : (invited paper)," *Int. Conf. on 6G Netw. (6GNet)*, 2022.

[16] D. V. Huynh, V. D. Nguyen, V. Sharma, O.A. Dobre, T. Q. Duong, "Digital twin empowered ultra-reliable and low-latency communications-based EDGE networks in industrial IOT environment," *IEEE Int. Conf. on Commun.*, 2022.

[17] 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. on Sel. Area. in Commun.*, vol. 38, no. 5, pp. 816–830, May 2020.

[18] Y. Wang, and Z. Jun, 'A Survey of Mobile Edge Computing for the Metaverse: Architectures, Applications, and Challenges." *Int. Conf. on Collaboration and Internet Computing (CIC)*, 2022.

[19] J. Zhou, D. Tian, Z. Sheng, X. Duan, and X. Shen, "Distributed task offloading optimization with queueing dynamics in multiagent mobile edge computing networks," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12311–12328, Aug. 2021.

[20] R. Lin et al., "Distributed optimization for computation offloading in edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8179–8194, Dec. 2020.

[21] 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.

[22] K. U. Latif, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong. "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities." *IEEE Commun. Surveys & Tutorials*, pp 1-1, 2022.

[23] T. Zhou, Y. Yue, D. Qin, X. Nie, X. Li, and C. Li, "Joint device association, resource allocation, and computation offloading in ultra-dense multi-device and multi-task IoT networks," *IEEE Internet Things J.*, early access, Mar. 23, 2022.

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

[25] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under the stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581.

[26] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190.

[27] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in MEC: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[28] M. H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," *IEEE Int. Conf. on Commun. (ICC)*, May 2016, pp. 1–6.

[29] J. Yan, S. Bi, Y. J. Zhang, M. Tao, "Optimal task offloading and resource allocation in Mobile-edge computing with inter-user task dependency," *IEEE Trans. on Wireless Commun.*, 19(1), pp. 235–250.

[30] J. Yan, S. Bi, and Y. A. Zhang. "Optimal offloading and resource allocation in mobile-edge computing with inter-user task dependency." *IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-8. IEEE, 2018.

[31] D. V. Huynh, V. Nguyen, S. R. Khosravirad, V. Sharma, O. A. Dobre, H. Shin, and T. Q. Duong, "URLLC Edge Networks With Joint Optimal User Association, Task Offloading, and Resource Allocation: A Digital Twin Approach," *IEEE Trans. on Commun.*, vol. 70, no. 11, pp. 7669–7682, Nov. 2022.

[32] Y. Chen, J. Zhao, J. Hu, S. Wan, and J. Huang, "Distributed Task Offloading and Resource Purchasing in NOMA-enabled Mobile Edge Computing: Hierarchical Game Theoretical Approaches," *ACM Trans. in Embedded Computing Systems*, May 2023.

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

[34] V. H., Dang, R. K. Saeed, M. Antonino, A. D. Octavia, and Q. D. Trung, "Edge intelligence-based ultra-reliable and low-latency communications for digital twin-enabled metaverse," *IEEE Wireless Commun. Lett.*, 11(8), pp. 1733–1737, 2022.

[35] V. H. Dang, V. Nguyen, S. R. Khosravirad, V. Sharma, O. A. Dobre, H. Shin, and Q. D. Trunk, "URLLC EDGE networks with Joint Optimal User Association, Task Offloading, and resource allocation: A Digital Twin Approach," *IEEE Trans. on Commun.*, 70(11), pp. 7669–7682, 2022.

[36] Q. Wu, J. Xu, Y. Zeng, D.W.K. Ng, N. Al-Dhahir, R. Schober, and A.L. Swindlehurst, "A Comprehensive Overview on 5G-and-Beyond Networks With UAVs: From Communications to Sensing and Intelligence", *IEEE J. on Sel. Area. in Commun.*, vol. 39, no. 10, pp. 2912-2945, 2021.