Digital Twin-enabled Low-Carbon Sustainable Edge Computing for Wireless Networks

Dang Van Huynh*, Saeed R. Khosravirad†, Vishal Sharma‡, Berk Canberk§, Octavia A. Dobre* and Trung Q. Duong*,‡
* Memorial University, Canada, e-mail:{vdhuynh, odobre, tduong}@mun.ca
† Nokia Bell Labs, e-mail: saeed.khosravirad@nokia-bell-labs.com
‡ Queen's University Belfast, UK, e-mail: v.sharma@qub.ac.uk
§ Edinburgh Napier University, UK, e-mail: b.canberk@napier.ac.uk

***Abstract*—The advancement of sophisticated communication technologies and robust computing systems has unlocked opportunities for new applications across various domains. While these applications promise enhanced convenience and improved living standards, they also raise a critical concern regarding the trade-off between convenience and environmental sustainability. This paper addresses this concern by investigating sustainable resource management, employing a digital twin approach to minimise CO_2 emissions in edge computing systems. Specifically, our aim is to reduce the amount of CO_2 emissions by optimising the allocation of computing and communication resources. This includes optimising transmit power, adjusting the clock speed for task processing, and making optimal decisions regarding task offloading. To tackle this complex optimisation problem, we employ an iteratively alternating optimisation algorithm. Through extensive simulations, we illustrate the efficacy of our proposed solution in not only mitigating CO_2 emissions but also optimising resource allocation, thereby contributing to both environmental sustainability and technological efficiency.**

I. Introduction

Unlike traditional methods that often involve sending data to distant data centers for processing, mobile edge computing (MEC) brings the computing power closer to where the data is generated [1]. This proximity results in faster processing times and quicker responses, making MEC particularly well-suited for applications that require immediate action, such as in factories, transportation systems, and immersive technologies [2], [3]. However, alongside its immense promise, MEC also presents technical challenges that warrant careful consideration and innovative solutions [4]. These challenges span various aspects of MEC implementation, including but not limited to, efficient resource management, seamless integration with existing infrastructure, and ensuring robust security protocols to safeguard sensitive data [5]. Among these challenges, the optimal design of joint communication and computing resource allocation stands out as a primary research focus [6]–[9].

In [6], a scheme for joint task offloading and resource allocation aimed at minimising total processing delay in Internet of vehicle (IoV) systems was proposed. The scheme optimised task scheduling, channel allocation, and computing resource allocation for the vehicles, aiming to enhance overall system efficiency. Additionally, a solution for distributed resource optimisation was introduced in [7] to address fairness-aware latency minimisation among users in MEC systems assisted by digital twin (DT) technology, which optimised various communication and computation variables, such as transmit power, bandwidth allocation, task offloading portions, and processing rates of user equipment, through both centralised and distributed optimisation approaches. Furthermore, the DT approach was also employed in [8] to model the computing capacity of physical MEC systems. An iterative algorithm was developed to optimise transmit power of IoT devices, user association, intelligent task offloading, and estimated CPU processing rates. Similarly, in [9], a DT framework for IoT networks was proposed, with unmanned aerial vehicles (UAVs) serving as flying MEC servers. These UAVs support on-the-fly task offloading, addressing end-to-end latency minimisation challenges. Overall, the research focus on resource allocation in MEC systems has garnered attention due to various technical challenges. However, there are still open issues to explore in these areas, including understanding the environmental impact of communication and computing systems, managing the trade-offs between system budget, computing capacity, quality-of-service (QoS), quality-of-experience (QoE), and reducing the carbon footprint released into the environment.

In the realm of sustainable computing, an emerging focus lies in the development of carbon-aware edge computing systems. These studies are dedicated to devising efficient solutions aimed at optimising resource management and minimising carbon footprint [10]–[14]. For instance, a sustainable resource management framework proposed in [10] leveraged deep reinforcement learning (DRL) models to reduce energy consumption and CO_2 emissions. Similarly, in [11], an optimal task scheduling and offloading solution was introduced to diminish carbon footprint within the realm of edge computing, employing a graph-based reformulation to tackle a mixed integer linear programming problem. Moreover, research delves into more intricate issues such as the joint machine learning (ML) task offloading and carbon emission rights purchasing problem, as discussed in [12]. Here, the two-timescale Lyapunov optimisation technique was utilised for optimal decision-making. Similarly, [13] proposed a DRL-based edge computing management strategy aimed at minimising long-term operational costs and promoting low-carbon edge computing. Furthermore, [14] presented a Lyapunov optimisation-based solution to reduce carbon emissions in computation-intensive tasks within queuing-aware network models. Overall, sustainable edge computing represents a significant research direction with direct implications for a global issue – reducing carbon footprint in advanced computing systems. Recent efforts in this area have been directed towards optimal designs of resource allocation solutions to minimise CO_2 emissions effectively.

Drawing from the preceding discussions, this paper presents a CO_2 emission-aware edge computing model, integrating DT technology. The primary aim is to minimise the maximum potential release of CO_2 emissions during the execution of computational tasks. The formulated problem accounts for crucial attributes inherent to MEC-based systems, including transmit power, clock speed of physical devices, energy budget of IoT devices, and task delay tolerance. Addressing this complex challenge is achieved through the implementation of an efficient alternating optimisation algorithm. Subsequent section presents the system model and the formulation of the optimisation problem.

II. System Model and Problem Formulation

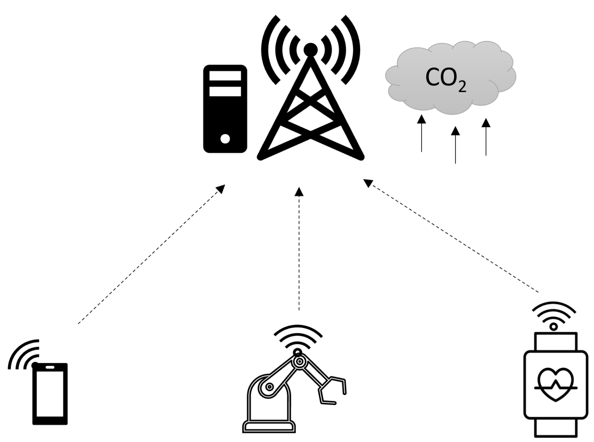

Fig. 1: An illustration of the MEC-based Internet of things systems.

In this paper, we consider a single base station, which is equipped with multiple antennas ($L > 0$) and associated with an edge server to process the ofloaded tasks from IoT devices. There are M single-antena IoT devices in the system denoted by the set of $\mathcal{M} = \{1, ..., M\}$. An illustration of the considered system model is displayed as Fig. 1. A computational task generated at the m-th IoT device is modelled as $J_m = (S_m, C_m, D_m^{\max})$, where S_m is the size of the task, C_m is the required CPU cycles to process the task, and $D_m^{\max}$ is the delay tolerance of the task. Due to limitations in terms of computing capacity and energy budget of constrained IoT devices, the m-th IoT device has to decide whether to offload the task to MEC or not, which is modelled as a binary variable $\alpha_m = \{0, 1\}$. Specially, when $\alpha_m = 1$, the computation task is ofloaded to the MEC; otherwise, it is processed locally at the m-th IoT devices.

A. Representations of DT-enabled Edge Computing Systems

DT has recently emerged as a promising technology to harness the potential of MEC systems, garnering significant attention from researchers in the field [8], [15], [16]. DT can be effectively utilised in MEC-based systems by virtually representing the computing capacity of physical devices, such as IoT devices and edge servers, enabling optimal decision-making for efficient control and management. In this paper, we adopt the assumption that DT is leveraged to comprehensively control and manage the entire physical system. To achieve this, the DT-enabled edge server is modelled as follows: $\mathrm{DT}_0 = (f_m^0, \hat{f}_m^0, \alpha_m)$, where f_m^0 (cycles/second) is the estimated clock speed of the edge server at DT, $\hat{f}_0$ is the deviation between the estimate and the real value of MEC's clock speed.

Regarding the m-th IoT device, its DT model is given by $\mathrm{DT}_m = (f_m, \hat{f}_m, p_m, \alpha_m)$, where $f_m, \hat{f}_m$ are the estimated and the deviation value of the clock speed of the IoT device, respectively. We assume that the necessary infrastructure is fully implemented to facilitate real-time interactions between the DT model and the physical system for data collection and remote control [17].

B. Wireless Transmission Model

Wireless connections between the base station (BS) and the m-th IoT device are established to facilitate communication. The channel vector $\mathbf{h}_m \in \mathbb{C}^{L\times 1}$ represents the connection between the BS and the m-th IoT device and is expressed as $\mathbf{h}_m = \sqrt{g_m}\bar{\mathbf{h}}m$, where g_m encompasses the large-scale channel coefficient, accounting for factors like pathloss and shadowing, while $\bar{\mathbf{h}}_m$ follows a small-scale fading distribution of $\mathcal{CN}(0, \mathbf{I})$. The received signal vector at the BS is represented as an $L \times 1$ matrix, expressed as

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{h}_m \sqrt{p_m} s_m + \mathbf{n}_k. \tag{1}$$

Here, p_m denotes the transmission power of the m-th device, s_m represents the zero-mean and unit-variance Gaussian information message from the m-th IoT device, and $\mathbf{n}_k \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_L)$ signifies the additive white Gaussian noise (AWGN) encountered during data transmission, where N_0 denotes the noise power. As a result, the transmission rate (bit/s) of wireless transmissions is calculated as

$$R_m(\mathbf{p}) = B\log_2\left(1 + \frac{p_m\|\mathbf{h}_m\|^2}{\mathcal{I}_m(\mathbf{p}) + N_0}\right) \tag{2}$$

where $\mathcal{I}_m(\mathbf{p}) = \sum_{n\neq m}^{M} \frac{p_n|\mathbf{h}_m^H\mathbf{h}_n|^2}{\|\mathbf{h}_m\|^2}$ is the interference power.

Then, the transmission delay (seconds) for task offloading from the m-th IoT device to the edge server is calculated as

$$D_m^{\mathrm{tx}}(\mathbf{p}) = \frac{\alpha_m S_m}{R_m(\mathbf{p})}. \tag{3}$$

C. Processing Delay Models

The processing delay of the computational task at the m-th IoT device is determined by the offloading decision, the required CPU cycles of the task, and the actual clock speed of the IoT device. This delay can be expressed as:

$$D_m^{\mathrm{IoT}}(\alpha_m, f_m) = \frac{(1-\alpha_m)C_m}{f_m + \hat{f}_m}. \quad (4)$$

Similarly, the processing delay of the task offloaded to the edge server is expressed as

$$D_m^{\mathrm{MEC}}(\alpha_m, f_m^0) = \frac{\alpha_m C_m}{f_m^0 + \hat{f}_m^0}. \quad (5)$$

Consequently, the end-to-end (e2e) delay of the offloaded task from the m-th IoT device consists of the local processing delay, wireless transmission delay, and the edge processing delay, defined as

$$D_m(\alpha_m, \mathbf{p}, \mathbf{f}) = D_m^{\mathrm{IoT}}(\alpha_m, f_m) + D_m^{\mathrm{tx}}(\mathbf{p}) + D_m^{\mathrm{MEC}}(\alpha_m, f_m^0). \quad (6)$$

D. Energy Consumption Models

To execute local processing and task offloading through wireless connections, IoT devices consume energy for these operations. Hence, the total energy consumption (measured in joules) of the m-th IoT device comprises the energy expended on local processing (E_m^{cp}) and the wireless transmission (E_m^{cm}), expressed as:

$$\begin{aligned} E_m(\alpha_m, \mathbf{p}, \mathbf{f}) &= E_m^{\mathsf{cp}}(\alpha_m, f_m) + E_m^{\mathsf{cm}}(\alpha_m, \mathbf{p}) \\ &= \theta_m(1-\alpha_m)C_m(f_m)^2 + \frac{p_m \alpha_m S_m}{R_m(\mathbf{p})}, \quad (7) \end{aligned}$$

where θ_m is the parameter for computation energy consumption of the m-th IoT (Watt.s^3/cycle^3).

Similarly, the energy consumption for computation at the MEC is given by

$$E_m^0(\alpha_m, \mathbf{f}) = \theta_0 \alpha_m C_m (f_m^0)^2, \quad (8)$$

where θ_0 is the parameter for computation energy consumption of the MEC server.

E. CO_2 Emissions Model

As IoT devices and edge servers actively participate in computing and communication tasks, the energy expended during these operations translates into CO_2 emissions, contributing to environmental impact. The environmental impact of computing and communication technologies is multifaceted, encompassing not only the direct emissions from energy usage but also the indirect impacts stemming from manufacturing, infrastructure, and electronic waste. Deriving the carbon emission amount involves navigating through various complex processes, yet a broad estimation can be attained by [10].

$$\xi_m(\alpha_m, \mathbf{p}, \mathbf{f}) = \eta C_{\mathrm{IE}} \Big[E_m(\alpha_m, \mathbf{p}, \mathbf{f}) + E_m^0(\alpha_m, \mathbf{f}) \Big], \quad (9)$$

where ξ_m is carbon emission (kg CO_2), C_{IE} is carbon intensity of electricity (kgCO_2/kWh), and $\eta = 2.77778e^{-7}$ used to convert energy consumption unit from Joule to kWh. The amount of C_{IE} differs across regions, and for the purposes of this paper, it stands at 182 gCO_2/kWh for the London area [10].

F. The Optimisation Problem

This paper focuses on minimising the maximum potential CO_2 emissions associated with processing computational tasks within the network, taking into account specified delay requirements, energy constraints for IoT devices, and the computing capacity of the edge server. Consequently, the optimisation problem addressed in this study is formulated as follows.

$$\begin{aligned} &\min_{\boldsymbol{\alpha}, \mathbf{p}, \mathbf{f}} \ \max_{\forall m} \{\xi_m(\boldsymbol{\alpha}, \mathbf{p}, \mathbf{f})\}, && (10a) \\ &\text{s.t. } D_m(\alpha_m, \mathbf{p}, \mathbf{f}) \le D_m^{\max}, \forall m && (10b) \\ &\quad E_m(\alpha_m, \mathbf{p}, \mathbf{f}) \le E_m^{\max}, \forall m, && (10c) \\ &\quad R_m(\mathbf{p}) \ge R_m^{\min}, \forall m, && (10d) \\ &\quad \sum_{m=1}^{M} \alpha_m f_m^0 \le F_{\max}, && (10e) \\ &\quad \alpha_m \in \{0, 1\}, \forall m. && (10f) \end{aligned}$$

As outlined in (10), constraint (10b) represents the delay requirement for each computational task. Constraints (10c) and (10e) define the energy budget allocated to IoT devices and the computing capacity available at the edge server, respectively. The quality of service (QoS) requirement for the wireless transmission link is specified in constraint (10d). Lastly, constraint (10f) pertains to the binary decision regarding task offloading.

III. Proposed Solutions

The problem (10) is evidently a mixed-integer nonlinear programming (MINLP) problem, posing significant computational challenges for direct solution. Compounding the complexity are the strong coupling between binary and continuous variables, exemplified by $E_m(\alpha_m, \mathbf{p}, \mathbf{f})$, $T_m(\alpha_m, \mathbf{p}, \mathbf{f})$, and non-convex constraints such as (10b) and (10c). To address these complexities, we introduce an alternating optimisation approach tailored to tackle this challenging problem [18]. In order to solve the problem (10) with the alternating approach, we consider three subproblems, including optimal transmit power control, optimal the estimated clock speed, and optimal task offloading decisions. The subsequent subsections detail the development of our proposed solution.

A. Optimal Transmit Power Control

To begin, we address the optimal transmit power problem. In order to find the most efficient power control for wireless transmissions from the m-th IoT devices to the BS, we establish the following optimisation problem.

$$\begin{aligned} &\min_{\mathbf{p} | \mathbf{f}^{(i)}, \boldsymbol{\alpha}^{(i)}} \ \max_{\forall m} \{\xi_m(\boldsymbol{\alpha}^{(i)}, \mathbf{p}, \mathbf{f}^{(i)})\}, && (11a) \\ &\text{s.t. } D_m(\alpha_m^{(i)}, \mathbf{p}, \mathbf{f}^{(i)}) \le D_m^{\max}, \forall m && (11b) \\ &\quad E_m(\alpha_m^{(i)}, \mathbf{p}, \mathbf{f}^{(i)}) \le E_m^{\max}, \forall m, && (11c) \\ &\quad R_m(\mathbf{p}) \ge R_m^{\min}, \forall m, && (11d) \end{aligned}$$

As we can see from (11), the objective function and constraints (11b), (11b), (11b) are non-convex. Therefore, we process these constraints by convexifying the transmission rate function, the delay and the energy expressions. Firstly, we address the non-convex transmission rate function because it is the main component of transmission latency as well as energy computation of communcations. To do this, we apply the following inequality [19]:

$$\ln(1+\frac{x}{y}) \geq u - \frac{v}{x} - wy, \tag{12}$$

where $u = \ln\left(1+\frac{\bar{x}}{\bar{y}}\right) + 2\frac{\bar{x}}{\bar{x}+\bar{y}} > 0$, $v = \frac{\bar{x}^2}{\bar{x}+\bar{y}} > 0$, and $w = \frac{\bar{x}}{(\bar{x}+\bar{y})\bar{y}} > 0$. with $x > 0, y >$, and $(\bar{x}, \bar{y})$ are the feasible point of (x, y) into (2) with $x = p_m\|\mathbf{h}_m\|^2$, $\bar{x} = p_m^{(i)}\|\mathbf{h}_m\|^2$, $y = \mathcal{I}_m(\mathbf{p}) + N_0$, and $\bar{y} = \mathcal{I}_m(\mathbf{p}^{(i)}) + N_0$. As a results, the transmission rate $R_m(\mathbf{p})$ can be approximated as follows

$$R_m(\mathbf{p}) \geq \frac{B}{\ln 2}\big[u - \frac{v}{x} - wy\big] \triangleq R_m^{(i)}(\mathbf{p}^{(i)}), \tag{13}$$

where

$$u = \ln\left(1+\frac{p_m^{(i)}\|\mathbf{h}_m\|^2}{\mathcal{I}_m(\mathbf{p}^{(i)})+N_0}\right) + \frac{2p_m^{(i)}\|\mathbf{h}_m\|^2}{p_m^{(i)}\|\mathbf{h}_m\|^2+\mathcal{I}_m(\mathbf{p}^{(i)})+N_0},$$
$$v = \frac{\big(p_m^{(i)}\|\mathbf{h}_m\|^2\big)^2}{p_m^{(i)}\|\mathbf{h}_m\|^2+\mathcal{I}_m(\mathbf{p}^{(i)})+N_0},$$
$$w = \frac{p_m^{(i)}\|\mathbf{h}_m\|^2}{(p_m^{(i)}\|\mathbf{h}_m\|^2+\mathcal{I}_m(\mathbf{p}^{(i)})+N_0\big)\big(\mathcal{I}_m(\mathbf{p}^{(i)})+N_0\big)}.$$

Consequently, the constraint (11d) is now equivalent to the following constraint

$$R_m^{(i)}(\mathbf{p}^{(i)}) \geq R_m^{\min}, \forall m, i. \tag{14}$$

To deal with (11b), we introduce variables $\mathbf{r} = \{r_m\}_{\forall m} \geq 1/R_m^{(i)}(\mathbf{p}^{(i)})$. At the i-th iteration, the constraint (11b) can be expressed as follows.

$$\frac{(1-\alpha_m^{(i)})C_m}{f_m^{(i)} - \hat{f}_m} + \alpha_m S_m r_m + \frac{\alpha_m^{(i)} C_m}{f_m^{0(i)} - \hat{f}_m^0} \leq D_m^{\max}, \tag{15}$$

which is a convex constraint under the variables of $\mathbf{p}$.

Similarly, we apply the introduced variables $\mathbf{r}$ to approximate $E_m(\alpha_m^{(i),\mathbf{p},\mathbf{f}^{(i)}})$ in (10c) as follows.

$$\theta_m(1-\alpha_m^{(i)})C_m(f_m^{(i)})^2 + p_m r_m \alpha_m^{(i)} S_m \leq E_m^{\max} \tag{16}$$

However, (16) is still not a convex constraint. Therefore, we apply the following inequality to convexify (16).

$$xy \leq \frac{1}{2}\left(\frac{\bar{y}}{\bar{x}}x^2 + \frac{\bar{x}}{\bar{y}}y^2,\right) \tag{17}$$

By substituting $x = p_m, \bar{x} = p_m^{(i)}, y = r_m, \bar{y} = r_m^{(i)}$, we can equivalently express (16) as the following convex constraint.

$$E_m^{(i)}(\boldsymbol{\alpha}^{(i)}, \mathbf{p}, \mathbf{f}^{(i)}) = \theta_m(1-\alpha_m^{(i)})C_m(f_m^{(i)})^2 + \frac{\alpha_m^{(i)} S_m}{2}\left(\frac{r_m^{(i)}}{p_m^{(i)}}p_m^2 + \frac{p_m^{(i)}}{r_m^{(i)}}r_m^2,\right) \leq E_m^{\max}. \tag{18}$$

Consequently, we have successfully transformed the problem (11) into a convex problem to solve at the i-th iteration as follows.

$$\min_{\mathbf{p}|\mathbf{f}^{(i)},\boldsymbol{\alpha}^{(i)}} \quad \max_{\forall m}\{\xi_m^{(i)}(\boldsymbol{\alpha}^{(i)}, \mathbf{p}, \mathbf{f}^{(i)})\}, \tag{19a}$$
$$\text{s.t. } (15), (18), (14), \tag{19b}$$

where $\xi_m^{(i)}(\boldsymbol{\alpha}^{(i)}, \mathbf{p}, \mathbf{f}^{(i)}) = E_m^{(i)}(\alpha_m, \mathbf{p}, \mathbf{f})\eta C_{\text{IE}}$. The problem is now can be solved efficiently with the well known CVX package.

B. Clock Speed Optimisation

The second subproblem solved in the alternating-based solution is the clock speed optimisation problem. In this subproblem, we solve for the optimal adjusting of the computing resource of the IoT devices and the edge server to execute the computational tasks. Given $(\mathbf{p}^{(i)}, \boldsymbol{\alpha}^{(i)})$, this subproblem finds the optimal clock speed, (i.e., $\mathbf{f}$). The optimisation problem is expressed as follows.

$$\min_{\mathbf{f}|\mathbf{p}^{(i)},\boldsymbol{\alpha}^{(i)}} \quad \max_{\forall m}\{\xi_m(\boldsymbol{\alpha}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f})\}, \tag{20a}$$
$$\text{s.t. } D_m(\boldsymbol{\alpha}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}) \leq D_m^{\max}, \forall m \tag{20b}$$
$$E_m(\boldsymbol{\alpha}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}) \leq E_m^{\max}, \forall m, \tag{20c}$$
$$(10e), (10f). \tag{20d}$$

As observed in problem (20), the energy consumption expression is quadratic in terms of $\mathbf{f}$, while the e2e delay expression is a combination of reciprocal functions involving the variables $\mathbf{f}$. Consequently, problem (20) is a convex program with respect to the variables $(f_m, f_m^0, \forall m)$, rendering it solvable using the CVX package.

C. Task Offloading Decisions Optimisation

Lastly, the optimisation of task offloading decisions addressed. Given $\mathbf{p}^{(i)}, \mathbf{f}^{(i)}$, this subproblem aims to identify the optimal task offloading decisions, represented by $\boldsymbol{\alpha}$. The optimisation problem can be expressed as follows.

$$\min_{\boldsymbol{\alpha}|\mathbf{p}^{(i)},\mathbf{f}^{(i)}} \quad \max_{\forall m}\{\xi_m(\boldsymbol{\alpha}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)})\}, \tag{21a}$$
$$\text{s.t. } D_m(\boldsymbol{\alpha}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)}) \leq D_m^{\max}, \forall m. \tag{21b}$$
$$E_m(\boldsymbol{\alpha}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}) \leq E_m^{\max}, \forall m, \tag{21c}$$
$$\sum_{m=1}^{M} \alpha_m f_m^{0(i)} \leq F_{\max}, \tag{21d}$$
$$(10f). \tag{21e}$$

This problem poses a mixed-integer (binary) programming challenge, known for its computational complexity. Fortunately, the MOSEK solver integrated into CVX is adept at handling such problems efficiently. Hence, we can determine optimal task offloading decisions by solving (21) with the given $(\mathbf{p}^{(i)}, \mathbf{f}^{(i)})$ parameters at the i-th iteration.

D. Proposed Alternating-based Algorithm

Building upon the aforementioned progress, we introduce an alternating-based optimisation algorithm designed to tackle (10), as outlined in Algorithm 1. The algorithm commences with an initialisation step, during which initial feasible points are derived using the formulations of the subproblems (19), (20), and (21). The sequence of the solving procedure is outlined as follows. First, the algorithm computes the optimal transmit power given the current values of the clock speed and offloading variables. Next, the clock speed optimisation is performed using the new transmit power solutions and the current offloading decisions. Finally, the algorithm determines the offloading decisions before commencing the next iteration. To ensure the feasibility of this initialization, we implement a validating function to verify that all constraints are satisfied with the selected parameters and initial points prior to the commencement of the first iteration.

Algorithm 1 : Proposed Algorithm for Solving (10).

1: **Initialisation**: Set $i = 1$, maximum number of iteration, $I_{\max}$; generate the initial feasible points, and choose the initial parameters for (10).
2: **while** (not Convergence or $i < I_{\max}$) **do**
3: Solve (19) with the given $\mathbf{f}^{(i)}, \boldsymbol{\alpha}^{(i)}$ for the next optimal solutions of the transmit power variables;
4: Solve (20) with the given $\mathbf{p}^{(i)}, \boldsymbol{\alpha}^{(i)}$ for the next optimal solutions of the clock speed variables;
5: Solve (21) with the given $\mathbf{p}^{(i)}, \mathbf{f}^{(i)}$ for the next optimal solutions of the offloading decision variables;
6: **end while**
7: **Solution**: optimal solutions of $\{\mathbf{p}^*, \mathbf{f}^*, \boldsymbol{\alpha}^*\}$.

IV. Simulation Results and Discussions

A. Simulation Setting

For simulations, we consider a small-scale IoT network, where all IoT devices are randomly distributed in a space of 100 m x 100 m. There are totally 6 devices connecting to the BS in the network. The large-scale fading for the wireless transmission from the m-th IoT device to the BS is modelled as $g_{mk} = 10^{\mathbf{PL}(\mathrm{d_{mk}})/10}$, where $\mathbf{PL}(d_{mk}) = -35.3 - 37.6 \log_{10} d_{mk}$ [19]. The single-sided noise spectral density is set to -174 dBm/Hz [19]. The task size is set to 1 MB and the delay tolerance of the task is set to 2 seconds. The maximum required CPU cycles of the task is set in the range of $[1500, 2000]$. Other parameters are provided in Table I. The simulations were conducted in MATLAB and the convex programs were solved by the CVX package [20].

TABLE I: Simulation Parameters [8], [13], [19].

Parameters	Value
Number of antennas	$L = 8$
Maximum transmit power	$P_m^{\max} = 23$ dBm
System bandwidth	$B = 10$ MHz
Maximum IoT clock speed	$F_m^{\max} = 2$ GHz
Computing capacity of edge server	$F_m^{\max} = 10$ GHz
Minimum data rate	$R^{\min} = 1$ Mbps
Maximum energy consumption	$E_m^{\max} = 1$ Joule
Effective capacitance coefficient	$\theta_m = 10^{-27}$ Watt.s^3/cycle^3
Carbon intensity of electricity	$C_{\mathrm{IE}} = 182$ g CO_2/kWh

B. Numerical Results and Discussions

In this subsection, we present the numerical results of the simulations to demonstrate the convergence pattern of the proposed algorithm and highlight the superior of the proposed solution in minimising the amount of CO_2 emissions.

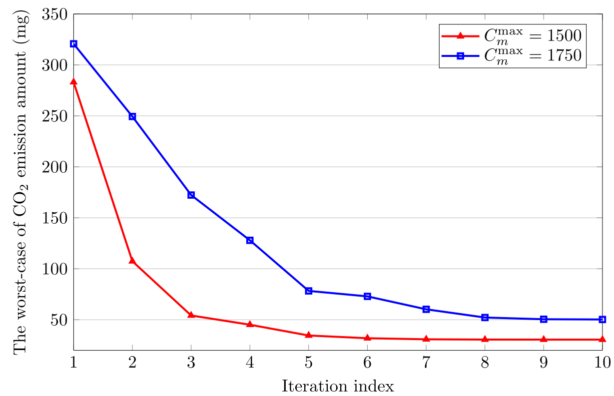

Fig. 2: Convergence pattern of Algorithm 1.

1) Convergence pattern of the proposed algorithm: To illustrate the convergence pattern of the proposed iterative algorithm, we monitor the worst-case CO_2 emissions among User Equipments (UEs) over the duration of its execution. As depicted in Figure 2, Algorithm 1 demonstrates its effectiveness in minimising CO_2 emissions, achieving a reduction of nearly 30 times in emissions after just 10 iterations. Notably, a significant reduction in emissions occurs after the initial iteration. This phenomenon can be attributed to the algorithm's ability to initiate optimisation from points that are considerably distant from the optimal solutions. Consequently, there is ample room for improvement during the first iteration. From the fifth iteration onwards, there is a gradual decrease in CO_2 emissions until convergence is reached. This observed trend underscores the algorithm's iterative refinement process, where adjustments are made iteratively to approach the optimal solution. This convergence behaviour emphasises the algorithm's potential to significantly enhance the sustainability of edge computing systems by efficiently managing resource allocation to minimise carbon emissions.

2) Effectiveness of the proposed solution: To highlight the efficacy of the proposed solution, we conduct a comparison between the results obtained from Algorithm 1 and those derived from a non-optimal benchmark scheme. Illustrated in Figure 3, our findings underscore the superior performance of the proposed solution in minimising CO_2 emissions. Specifically, under identical simulation conditions, the results obtained from Algorithm 1 exhibit a remarkable reduction, with CO_2 emissions amounting to less than 10 milligrams. This substantial improvement can be attributed to the holistic optimisation approach employed by Algorithm 1, which jointly addresses parameters such as transmit power, clock speeds of both IoT devices and edge servers, and task offloading decisions. Furthermore, Figure 3 also sheds light on the relationship between the number of CPU cycles required to

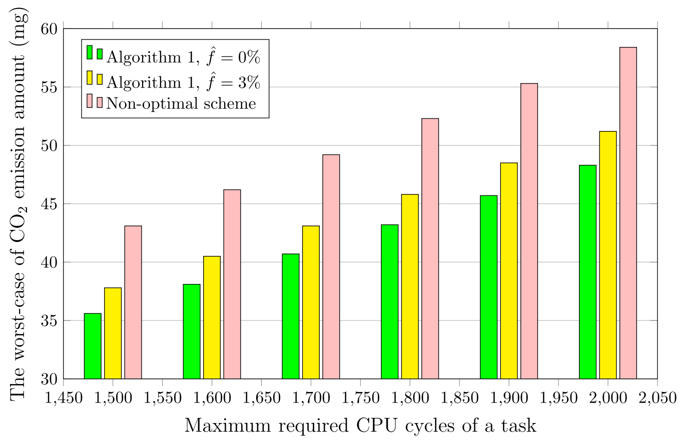

Fig. 3: Effectiveness of the proposed solution in minimising the amount of CO_2 emission.

execute a task and the resultant CO_2 emissions. Evidently, tasks demanding higher CPU cycles correspond to increased CO_2 emissions, highlighting the direct impact of computational intensity on environmental footprint. Moreover, our comparison in Figure 3 also illustrates the influence of DT estimation accuracy on the achieved results. Notably, the more precise the DT estimation of clock speeds, the lower the CO_2 emissions recorded. This observation indicates the pivotal role of accurate DT models and latency estimation in optimising resource allocation strategies and minimising environmental impact.

V. Conclusion

In summary, this paper has examined sustainable resource management within edge computing systems, employing the DT approach. The main objective has been to reduce CO_2 emissions in common MEC-aided IoT scenarios by optimising several factors, such as the IoT device transmit power, the DT estimated clock speeds, and the task offloading decisions. To tackle the complex challenge, the alternating optimisation algorithm was carefully developed. Through extensive simulations, the effectiveness of the proposed solution has been validated, demonstrating significant reductions in CO_2 emissions and optimised resource allocation. This confirmation highlights the potential of the DT-based approach to enhance sustainability within edge computing environments. Looking forward, there is an exciting opportunity for further research, particularly in developing practical machine learning solutions tailored for large-scale networks. By leveraging advanced algorithms and data-driven approaches, future efforts can aim to meet the evolving needs of modern computing systems while minimising their environmental impact.

Acknowledgement

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109 and in part by the Canada Research Chair Program CRC-2022-00187. The work of B. Canberk is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) 1515 Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab, Project 5239903.

References

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

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

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

[4] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug. 2019.

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

[6] W. Fan, Y. Su, J. Liu, S. Li, W. Huang, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for vehicular edge computing based on V2I and V2V modes," *IEEE Trans. on Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4277–4292, Apr. 2023.

[7] D. V. Huynh, V.-D. Nguyen, S. R. Khosravirad, K. Wang, G. K. Karagiannidis, and T. Q. Duong, "Distributed communication and computation resource management for digital twin-aided edge computing with short-packet communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3008–3021, Aug. 2023.

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

[9] Y. Li, D. V. Huynh, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, "Unmanned aerial vehicles-aided edge networks with ultra-reliable low-latency communications: A digital twin approach," *IET Signal Processing*, Apr. 2022.

[10] M. Golec, S. S. Gill, F. Cuadrado, A. K. Parlikad, M. Xu, H. Wu, and S. Uhlig, "ATOM: Ai-powered sustainable resource management for serverless edge computing environments," *IEEE Trans. on Sustainable Comput.*, pp. 1–13, 2023.

[11] Z. Yu, Y. Zhao, T. Deng, L. You, and D. Yuan, "Less carbon footprint in edge computing by joint task offloading and energy sharing," *IEEE Netw. Lett.*, vol. 5, no. 4, pp. 245–249, Dec. 2023.

[12] H. Ma, Z. Zhou, X. Zhang, and X. Chen, "Toward carbon-neutral edge computing: Greening edge ai by harnessing spot and future carbon markets," *IEEE Internet of Things J.*, vol. 10, no. 18, pp. 16 637–16 649, Sep. 2023.

[13] L. Gu, W. Zhang, Z. Wang, D. Zeng, and H. Jin, "Service management and energy scheduling toward low-carbon edge computing," *IEEE Trans. on Sustainable Comput.*, vol. 8, no. 1, pp. 109–119, 2023.

[14] C.-S. Yang, C.-C. Huang-Fu, and I.-K. Fu, "Carbon-neutralized task scheduling for green computing networks," in *Proc. 2021 IEEE Glob. Commun. Conf.*, Rio de Janeiro, Brazil, Dec. 2022, pp. 4824–4829.

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

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

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

[18] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *AFSS international conference on fuzzy systems.* Springer, 2002, pp. 288–300.

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

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