A fundamental problem in MEC is the scheduling of offloading tasks, which poses great challenges. For example, due to the limited battery capacity and computation resource, mobile users usually offload their computing-intensive tasks to MEC servers to prolong the battery life of their mobile devices. To meet stringent delay requirement of an offloaded task for delay-sensitive applications such as interactive gaming and augmented reality, each offloaded task needs to be partitioned into multiple chunks and be offloaded to different servers for processing. It becomes critical to optimally allocate tasks from mobile devices to MEC servers in order to minimize the energy consumption of mobile devices while meeting delay requirements of mobile users. Furthermore, when a task is executed on a mobile device, the amount of energy consumed at the mobile device can be minimized by optimally scheduling the CPU-cycle frequency of the mobile device via Dynamic Voltage and Frequency Scaling (DVFS) \cite{18}. It is worth noting that task allocation needs to be considered with the CPU-cycle frequency scheduling. When performing a task allocation, which size of part of the task should be processed locally, i.e., the number of chunks of the task to be executed on its local mobile device, needs to be determined too. In this paper we will address the mentioned challenges.

The novelty of this paper lies in the formulation of a delay-energy joint optimization problem in MEC and a novel solution to the problem is provided through a series of reductions.

The main contributions of this paper are summarized as follows.

- We study task offloading from mobile devices to MEC. We formulate a novel delay–energy joint optimization problem for task offloading in MEC that jointly takes into account both response delays and energy consumptions on mobile devices.
- We first formulate a mixed-integer nonlinear program for the problem, we then relax the problem into a nonlinear program problem that can be solved in polynomial time. We later show how to derive a feasible solution to the problem from the solution of the relaxed problem.
- We finally evaluate the performance of the proposed algorithm for the joint optimization problem through experimental simulation. Experimental results demonstrated the proposed algorithm is very promising, and outperforms its analytical solution as the theoretical estimation is conservative.

The rest of this paper is organized as follows. Section II introduces...
roduces the network model and problem formulation. Section III provides a solution to the relaxed version of the problem, and Section IV shows how to derive a feasible solution to the original problem for the solution to the relaxed one. Section V evaluates the performance of the proposed algorithm through experimental simulations. Section VI surveys related works, and Section VII concludes the paper.

II. PRELIMINARIES

In this section, we first introduce the network model. We then introduce the local computing model at each mobile device, including the local execution delay and the energy consumption. We finally introduce the mobile-edge computing model that includes the energy consumption of offloading tasks to MEC through wireless transmission and the processing delay of offloaded tasks in MEC.

A. Network model

We consider a mobile-edge computing network (MEC) that consists of numbers of Access Points (AP) and servers. Denote by \( S = \{s_1, s_2, \cdots, s_N\} \) the set of the servers in the network and \( N \) is the number of servers in the MEC, as shown in Fig. 1. For the sake convenience, we here consider a single mobile device (MD) who accesses a nearby AP to offload his tasks for processing, and we further assume that there is neither wireless channel interferences at the AP nor overloading issues on the servers. We adopt the similar assumptions imposed on each offloading task [14], that is, each task can be arbitrarily divided into different-sized chunks and processed at different servers and/or local mobile devices. We assume that the global information of servers in MEC is given at the AP.

Fig. 1: An illustrative example of a MEC network with one task offloading from a mobile user.

For a given task, whether it is offloaded to MEC entirely or how much proportion of the task will be processed by its mobile device locally, it will determined by a central system scheduler through the AP, which is responsible for the decision and distribution of task chunks to different servers in MEC.

B. Local computing at the mobile device

Let \( A(L, \tau_d) \) denote a task, where \( L \) (in bits) is the input size of the task and \( \tau_d \) is the execution deadline, i.e., if the task is to be executed, it should be finished within \( \tau_d \) time units. For convenience, denote by \( s_0 \) the mobile device of the task. Let \( z_i \) be a binary variable, where \( z_i = 1 \) if the task is partially offloaded to server \( s_i \) and \( z_i = 0 \) otherwise for all \( i \) with \( 1 \leq i \leq N \), and \( z_0 = 1 \) if there is part of the task to be processed at the mobile device. Denote by \( \eta_i \) the percentage of task \( A(L, \tau_d) \) offloaded to server \( s_i \) with \( 0 \leq \eta_i \leq 1 \). Then, we have

\[
\sum_{i=0}^{N} z_i \eta_i = 1, \quad 0 \leq i \leq N.
\]

A mobile device can execute its task locally. This incurs the local execution delay and energy consumption at the mobile device, while they are jointly determined by the number of CPU cycles of the mobile device. Denote by \( \gamma_A \) the number of CPU cycles required for processing one bit of task \( A(L, \tau_d) \), which usually is given and can be obtained through off-line measurements [17]. Hence, the number of CPU cycles needed for processing local chunk of task \( A(L, \tau_d) \) is \( W = \gamma_A \sum_{i=0}^{N} z_i \eta_i \). Since the CPU-cycle frequencies of the mobile device are adjustable through DVFS techniques to reduce the energy consumption, denote by \( f_w \) the CPU frequency of the mobile device at the \( w \)th CPU cycle with \( w = 1, \cdots, W \), which is constrained by \( f_{\text{max}} \) as follows.

\[
0 \leq f_w \leq f_{\text{max}}.
\]

As a result, the local execution delay \( D_l \) of \( A(L, \tau_d) \) is defined as

\[
D_l = \sum_{w=1}^{W} (f_w)^{-1},
\]

and the energy consumption \( E_l \) of the local execution of \( A(L, \tau_d) \) is

\[
E_l = \kappa \sum_{w=1}^{W} f_w^2,
\]

where the constant \( \kappa \) is the effective switched capacitance that depends on the chip architecture [13].

C. Server computing in MEC

The mobile device offloads its tasks to MEC via the AP. Such task offloading involves two important metrics: the \textit{end-to-end delay} between the time point that a mobile device offloads its task to the MEC and the time point that the execution result of the offloaded task received by the mobile device; and the \textit{energy consumption} of the mobile device on its transmission of the task chunks from the mobile device to the AP. Specifically, the end-to-end delay consists of the transmission delay from the mobile device to the AP; the transmission delay of task chunks of \( A(L, \tau_d) \) from the AP to the servers; and the processing delay of task chunks in servers. In the following we give the detailed definition of these two metrics.

Let \( \tau_{h,p} \) denote the maximum achievable uplink transmission rate between the mobile device and the AP which is given in advance. Then, the transmission delay \( D_{lp} \) of task chunks of \( A(L, \tau_d) \) from the mobile device to the AP is

\[
D_{lp} = \frac{(1 - z_0 \eta_0) L}{\tau_{h,p}},
\]

where \( z_0 \eta_0 \) is the percentage of \( A(L, \tau_d) \) processed locally.

The task chunks then are distributed to different servers in MEC once the mobile device finishes its task transmission. We

\* \( W \) is truncated to a nearest integer \( \lceil W \rceil \) if it is not an integer.
We then define the amount of energy consumed \( E \) (i.e., CPU cycles per second). The processing delay \( D \) execution.

A task offloading of mobile device will continue to hold the channel for a while transmitting the task chunks of mobile device to each chosen server \( s \).

Denote by \( R \) the transmission rate between the AP and \( s \). Then, the transmission delay of a subtask in server \( s \) is

\[
R = \left\{ \begin{array}{ll}
\max & \text{if } d > 0, \\
0 & \text{otherwise.}
\end{array} \right.
\]

Notice that the optimal solution to \( P \) is relaxed into a nonlinear program \( P' \) (Section IV A), and an algorithm is then devised for problem \( P_0 \) in Section IV.B.

### B. Problem relaxation

\( P_0 \) includes integer variables \( z_i \) and real number variables \( \eta_i \) and \( f_w \). We substitute \( z_i \eta_i \) with \( x_i \) to eliminate integer variables and drop the tail energy \( E_t \) in \( P_0 \).

Denote by \( \mathcal{S} = \{ s_{j1}, \ldots, s_{jn} \} \subset \mathcal{S} \) the set of \( n \) selected servers for task \( A(L, \tau_d) \) while the execution of the offloaded subtasks in MEC meets the overall delay of the task, where \( 0 < n \leq m \), \( z_{ji} = 1 \), and \( i = 1, \ldots, n \). Hence, Having determined the chosen servers to allocate the subtasks of Task \( A(L, \tau_d) \), problem \( P_0 \) now can be relaxed into a nonlinear program \( P_1 \) as follows.

\[
P_1 : \min \quad \kappa \sum_{w=1}^{w_0} f_w^2 - \phi x + \alpha \max \{ D, R \}
\]

\[
\text{s.t. } \left\{ \begin{array}{l}
\sum_{w=1}^{w_0} x_i = 1, \\
x_i \geq 0, \forall i = 0, 1, \ldots, n,
\end{array} \right.
\]

\[
R = \max \left\{ \sum_{1 \leq i \leq n} R_i \right\}
\]

\[1\] The CPU-cycle frequencies are integers. While \( f_w \) is a very large integer, often near to \( 10^7 \). Hence, we can truncate the real number value of \( f_w \) to an integer, which hardly impacts the analysis.
where $x_i$ and $f_w$ are variables. Constraints (13) and (14) in $P_1$ are equivalent to Constraints (1), (10) and (11) in $P_0$. The difference between the objectives of $P_0$ and $P_1$ is $\phi = E_i$.

Recall that $z_j, c_j, \gamma_A$, and $r_{h,p}$ are constants.

We assume $\sum_{u=1}^{B_0x_0} f_w^2 > P_{tx}x_0L/r_{h,p}$ in this paper, i.e., the amount of energy consumed for executing a task with size $x_0L$ at the mobile device is greater than that for transmitting the task chunks from the mobile device to the AP without considering the tail energy. Hence, the value of Problem $P_1$ is greater than 0.

**C. Solution to the relaxed problem $P_1$**

To solve problem $P_1$, we distinguish it into two cases: Case 1. $D_t \geq R_{\text{max}}$; and Case 2. $D_t \leq R_{\text{max}}$. We adopt a strategy of solving $P_1$ through the transformation of problem $P_1$ into another nonlinear program without the variable $f_w$, i.e., the CPU-cycle frequency. We then solve the nonlinear program as follows.

**Case 1.** The local execution delay $D_t$ is no less than the end-to-end delay $R_{\text{max}}$, and thus determines the overall delay. Problem $P_1$ then can be rewritten as the following problem $P_2$.

$$P_2: \begin{array}{ll}
\min & k \sum_{u=1}^{B_0x_0} f_w^2 - \phi x_0 + \alpha \sum_{u=1}^{B_0x_0} (f_w)^{-1} \\
\text{s.t.} & \sum_{u=1}^{B_0x_0} f_w^{-1} \geq R_{\text{max}}, \\
& R_{\text{max}} = \max_{1 \leq i \leq n} \{R_i\}.
\end{array} \quad (23)$$

In the following we solve problem $P_2$ by first solving its simplified version that focuses on CPU frequencies, and then extending this simplified solution to solve $P_2$.

Let $B = B_0x_0$, and let $F(f_1, \cdots, f_B)$ be the sum of the first and third terms in the optimization objective of $P_2$ that includes CPU-frequency variable $f_w$ only, i.e.,

$$F(f_1, \cdots, f_B) = k \sum_{u=1}^{B} f_w^2 + \alpha \sum_{u=1}^{B} (f_w)^{-1}. \quad (25)$$

The partial derivative of $F(f_1, \cdots, f_B)$ with respect to $f_w$ is

$$\frac{\partial F}{\partial f_w} = 2k f_w - \alpha (f_w)^{-2}, \quad (26)$$

where $w = 1, \cdots, B$.

If $\alpha/(2k)^{1/3} \leq f_{\text{max}}$, the only stationary point of $F(f_1, \cdots, f_B)$ is $f_w = \alpha/(2k)^{1/3}$ for all $w$ with $1 \leq w \leq B$, and the minimum value of $F(f_1, \cdots, f_B)$ is $\kappa B(\alpha/2k)^{2} + \alpha B(\alpha/2k)^{-\frac{1}{3}}$. Otherwise, we have $\frac{\partial F}{\partial f_w} < 0$, and $F(f_1, \cdots, f_B)$ reaches the minimum value of $\kappa B f_{\text{max}}^2 + \alpha B f_{\text{max}}^{-\frac{1}{3}}$ when $f_w = f_{\text{max}}$ for all $w$ with $1 \leq w \leq B$.

In summary, to achieve the minimum value of $F(f_1, \cdots, f_B)$, each $f_w$ with $1 \leq w \leq B$ must be set as follows.

$$f_w = \bar{f} = \begin{cases} \left(\frac{\alpha}{2k}\right)^{\frac{2}{3}}, & \text{if } \left(\frac{\alpha}{2k}\right)^{\frac{2}{3}} \leq f_{\text{max}} \\ f_{\text{max}}, & \text{otherwise}. \end{cases} \quad (27)$$

Having the solution to $F(f_1, \cdots, f_B)$, we now solve problem $P_2$ whose objective is the sum of term $-\phi x_0$ and $F(f_1, \cdots, f_B)$. Notice that (i) $F(f_1, \cdots, f_B)$ is a function of variables $x_0$ and $f_w$, while item $-\phi x_0$ is the function of variable $x_0$ only. Thus, the objective function of $P_2$ is the sum of functions $F(f_1, \cdots, f_B)$ and $-\phi x_0$. Before we solve $P_2$, we present the following lemma.

**Lemma 1.** Given three functions $g_1(x, y)$, $g_2(x)$ and $g_3(x, y) = g_1(x, y) + g_2(x)$, where $x$ and $y$ are both sets, then we have that

$$\min_{x, y} g_3(x, y) = \min_x \{\min_y g_1(x, y) + g_2(x)\}.$$

**Proof.** The proof can be obtained by contradiction. Let $\{x^*, y^*\}$ be the optimal solution for $\min g_3(x, y)$. Assume that $y^*$ is not the optimal solution for $\min y g_1(x^*, y)$. Then there must exist a $y$ satisfying $g_3(x^*, y^*) - g_3(x^*, y) = g_1(x^*, y^*) - g_1(x^*, y) > 0$, which contradicts that $g_3(x^*, y^*)$ is the minimum. Therefore, the assumption is not true and $y^*$ is the optimal solution for $\min_y g_1(x^*, y)$, i.e., the optimal solution for $\min g_3(x, y)$ should be in

$$\{(x, y^*) \mid y^* = \arg\min_y g_1(x, y)\}.$$

Lemma 1 then follows. \hfill \blacksquare

Following Lemma 1 we rewrite the optimization objective of $P_2$ by replacing its terms $\kappa \sum_{u=1}^{B} f_w^2 + \alpha \sum_{u=1}^{B} (f_w)^{-1}$ with $B(\kappa f_w^{\frac{2}{3}} + \alpha (f_w)^{-\frac{1}{3}})$. We term this transformed optimization problem as problem $P_3$, which is defined as follows.

$$P_3: \begin{array}{ll}
\min & k B_0f^2 + \alpha B_0f - \phi x_0 \\
\text{s.t.} & (8), (13), (14), (18), \text{and} \ (19) \\
& B_0 = \gamma_A L \\
& q_0(1 - x_0) + q_i x_i \leq \frac{B_0}{f} x_0. \quad (28)
\end{array}$$

It can be seen that $P_3$ is a linear program, which can be solved in polynomial time.

**Case 2.** If the local executing delay $D_t$ is no greater than the end-to-end delay $R_{\text{max}}$, problem $P_1$ can be rewritten as problem $P_4$ as follows.

$$P_4: \begin{array}{ll}
\min & k \sum_{u=1}^{B_0x_0} f_w^2 - \phi x_0 + \alpha R_{\text{max}} \\
\text{s.t.} & (2), (3), (8), (13), (14) \text{ and } (24) \\
& \sum_{w=1}^{B_0x_0} f_w^{-1} \leq R_{\text{max}}. \quad (29)
\end{array}$$
We adopt the similar strategy for solving $P_3$ as we did for $P_2$. It can be seen the objective function of $P_2$ consists of two functions: function $\kappa \sum_{w=1}^{B_0x_0} f_w^2$ has variables $f_w$ and $x_0$; and function $-\phi x_0 + \alpha R_{\text{max}}$ has variables $x_i$ as $R_{\text{max}}$ is a function of $x_i$ with $0 \leq i \leq n$.

By Lemma 1 we first determine the value of each $f_w$ to minimize the value of function $\kappa \sum_{w=1}^{B_0x_0} f_w^2$ as follows.

By Inequalities (2) and (29), we have

$$B_{f_{\text{max}}} \leq \frac{1}{B_{f_{\text{min}}}} \leq R_{\text{max}}. \quad (30)$$

By Inequality (30), we have

$$\frac{B}{R_{\text{max}}} \leq f_{\text{max}}. \quad (31)$$

Denote by $R' = \frac{B}{R_{\text{max}}} (1/\sum_{w} f_w)$. Then, $R' \leq R_{\text{max}}$ by Inequality (29).

By Jensen’s inequality (see the appendix), we then have

$$\sum_{w=1}^{B} f_w^2 \geq \frac{B^3}{R^2} \geq \left(\sum_{w=1}^{B} f_w \right)^2. \quad (32)$$

Hence, $\sum_{w=1}^{B} f_w^2$ reaches the minimum $B^3/R^2$ when $f_w = \frac{B_{f_{\text{max}}}}{R_{\text{max}}} \sum_{w=1}^{B} f_w$ for all $w$ with $1 \leq w \leq B$. Then, $D_i = R_{\text{max}} = \sum_{w=1}^{B} f_w$.

Problem $P_5$ is then derived as follows, by replacing $\sum_{w=1}^{B} f_w^2$ in $P_4$ with $B^3/R^2$.

$$P_5: \quad \min_{x_i, y_i, i} \frac{Kx_0^3}{R_{\text{max}}} - \phi x_0 + \alpha R_{\text{max}}$$

s.t. (5) - (8), (13), (14), (22) and (24).

$$K = \kappa B_0^3. \quad (33)$$

Lemma 2. $P_3$ is equivalent to $P_5$.

Proof. It can be seen that problem $P_3$ is a special case of problem $P_5$ when $R_{\text{max}} = (B_0x_0)/f_{\text{min}}$. Under Case 1, $P_1$ is equivalent to $P_5$; while under Case 2, $P_1$ is equivalent to $P_5$. Hence, $P_1$ is equivalent to $P_5$.

The rest is to solve the nonlinear program problem – $P_5$. Before we proceed, we have the following lemma.

Lemma 3. If $f(\cdot)$ is a real-valued function of two variables of $x$ and $y$, which is defined whenever $x \in X$ and $y \in Y$ and $X$ and $Y$ are two domains. Then

$$\min_{x \in X, y \in Y} f(x, y) = \min_{x \in X} \min_{y \in Y} f(x, y). \quad (34)$$

Proof. We show the claim by contradiction. It can be seen that

$$\min_{x \in X, y \in Y} f(x, y) \leq \min_{x \in X} \min_{y \in Y} f(x, y).$$

Assuming that $(x^*, y^*)$ is the optimal solution of $\min_{x \in X, y \in Y} f(x, y)$. That is,

$$f(x^*, y^*) = \min_{x \in X, y \in Y} f(x, y). \quad (35)$$

Suppose the claim (34) does not hold, and we assume that

$$\min_{x \in X, y \in Y} f(x, y) < \min_{x \in X} \min_{y \in Y} f(x, y).$$

By Ineq. (36), we have

$$f(x^*, y^*) < \min_{y \in Y} f(x^*, y).$$

However, by Eq. (35), we have

$$f(x^*, y^*) \geq \min_{y \in Y} f(x^*, y).$$

This leads to a contradiction. The lemma thus follows. ■

It can be seen that the objective function of $P_5$ contains variables $x_0$ and $R_{\text{max}}$. Denote by $H(x_0, R_{\text{max}})$ the objective function of $P_5$ which is defined as follows.

$$H(x_0, R_{\text{max}}) = \frac{Kx_0^3}{R_{\text{max}}} - \phi x_0 + \alpha R_{\text{max}}. \quad (36)$$

To minimize $H(x_0, R_{\text{max}})$, we first determine the range of $R_{\text{max}}$ with respect to $x_0$ in order to meet Eq. (24). We then transform problem $P_5$ into a problem with respect to $x_0$ only, by adopting Lemma 3. The range of $R_{\text{max}}$ is determined by the following lemma.

Lemma 4. The range of $R_{\text{max}}$ with respect to $x_0$ is

$$(1 - x_0)Q \leq R_{\text{max}} \leq (1 - x_0)Q_u \quad (37)$$

where

$$Q = \sum_{i=1}^{n} 1/q_i \quad (38)$$

$$\bar{Q} = q_0 + 1/Q, \quad (39)$$

$$Q_u = q_0 + \max_{1 \leq i \leq n} \{q_i\} \quad (40)$$

$R_{\text{max}}$ reaches the minimum $\bar{Q}(1 - x_0)$ when

$$x_i = \frac{1 - x_0}{Q/q_i} \quad \text{for all } i \text{ with } 1 \leq i \leq n. \quad (41)$$

Proof. Recall that $R_{\text{max}} = \max_{1 \leq i \leq n} (R_i)$ by Eq. (24). Determining the range of $R_{\text{max}}$ is equivalent to determining the upper and lower bounds of $\max_{1 \leq i \leq n} (R_i) = \max_{1 \leq i \leq n} \{q_i x_i + q_0(1 - x_0)\}$, subject to that $\sum_{i=1}^{n} x_i = 1 - x_0$ with $x_i \geq 0$.

We first derive the upper bound of $R_{\text{max}}$. As $x_i \leq 1 - x_0$ and $q_i \leq \max_{1 \leq i \leq n} \{q_i\}$, we have $\max_{1 \leq i \leq n} \{q_i x_i\} \leq \max_{1 \leq i \leq n} \{q_0(1 - x_0)\}$. Hence, $R_{\text{max}} \leq (1 - x_0)Q_u$ holds.

We then derive the lower bound of $R_{\text{max}}$, i.e.,

$$\min_{1 \leq i \leq n} \{q_i x_i + q_0(1 - x_0)\} \geq q_0(1 - x_0).$$

As term $q_0(1 - x_0)$ is independent of $R_{\text{max}}$, we will focus on term $q_i x_i$. In the following we show that $\min_{1 \leq i \leq n} \{q_i x_i\}$ reaches the minimum $(1 - x_0)/Q$ when $x_i = \frac{1 - x_0}{Q/q_i}$ by contradiction.

Without loss of generality, we suppose that the optimal solution of $\min_{1 \leq i \leq n} \{q_i x_i\}$ satisfies that $q_1 x_1 \leq q_2 x_2 \leq \cdots \leq q_n x_n$. Then, $q_n x_n - q_1 x_1 = 0$. We now show that $q_n x_n - q_1 x_1 = 0$ by contradiction. Assume that $q_n x_n - q_1 x_1 = \epsilon > 0$, we then can have another feasible solution for $\min_{1 \leq i \leq n} \{q_i x_i\}$ $x' = \{x_1', x_2', \cdots, x_{n-1}', x_n\}$, where

$$x_1' = x_1 + \frac{\epsilon}{x_1 + x_n},$$

$$x_i' = x_i, \quad 2 \leq i \leq n - 1.$$
\[ x_n = x_n - \frac{\epsilon}{x_1 + x_n}. \]

Since \( q_n x_n < q_n x_n \), the value of \( \min \max_{1 \leq i \leq n} \{ q_i x_i \} \) under \( \mathcal{X} \) is \( \min \{ q_n x_n - 1, q_n x_n \} \leq q_n x_n \), \( \mathcal{X} \) is better than \( \mathcal{X} \). This leads to a contradiction that \( \mathcal{X} \) is the optimal solution to the problem. Therefore,

\[ q_1 x_1 = q_2 x_2 = \cdots = q_n x_n. \]

That is, \( \min \max_{1 \leq i \leq n} \{ q_i x_i \} \) reaches the minimum \((1 - x_0)/\bar{Q} \) when Eq. (41) holds. We thus have \( R_{\max} \geq (1 - x_0)/\bar{Q} + q_0 (1 - x_0) = (1 - x_0) \bar{Q} \), thereby \((1 - x_0) \bar{Q} \leq R_{\max} \). Inequalities (37) thus hold.

Since \( R_{\max} \) is a function of \( x_0 \), \( \min_{R_{\max}} H(x_0, R_{\max}) \) is also a function of \( x_0 \) too.

**Lemma 5.** \( \min_{R_{\max}} H(x_0, R_{\max}) \) is equivalent to function \( h(x) \), where \( h(x) \) is defined as follows.

\[
h(x) = \begin{cases} 
    h_1(x), & x \in \left[ 0, \frac{Q}{R + \bar{Q}} \right] \\
    h_2(x), & x \in \left( \frac{Q}{R + \bar{Q}}, 1 \right] \\
    h_3(x), & x \in \left( \frac{Q}{R + \bar{Q}}, \frac{Q}{R + \bar{Q}} \right]
  \end{cases}
\]  

where\[
  h_1(x) = \frac{K}{Q^2} x^3 - \phi x + \alpha (1 - x) \bar{Q}, \\
  h_2(x) = \frac{K}{Q u} x^3 - \phi x + \alpha (1 - x) Q u, \\
  h_3(x) = \left( \frac{K}{R^2} - \phi + \alpha R^* \right) x, \\
  R^* = \left( \frac{2K}{\alpha} \right)^{\frac{3}{2}}.
\]  

**Proof.** The partial derivative of \( H(x_0, R_{\max}) \) with respect to \( R_{\max} \) is \( \frac{\partial H(x_0, R_{\max})}{\partial R_{\max}} = -2K x_0^3 + \alpha \). Let \( \frac{\partial H(x_0, R_{\max})}{\partial R_{\max}} = 0 \), its solution is \( R^* x_0 \). If \( R^* x_0 \geq (1 - x_0) Q u \), i.e., \( \frac{Q u}{R + \bar{Q}} \leq x_0 \leq 1 \), \( H(x_0, R_{\max}) \) is a monotonically decreasing function. Hence, we have \( \min_{R_{\max}} H(x_0, R_{\max}) = H(x_0, (1 - x_0) Q u) \), which can be rewritten as \( h_3(x) \). Similarly, we can obtain the results for \( R^* x_0 \leq (1 - x_0) \bar{Q} \) and \( (1 - x_0) \bar{Q} \leq R^* x_0 \leq (1 - x_0) Q u \).}

**Theorem 1.** The optimal solution to problem \( P_1 \) is \( OPT(P_1^{\mathcal{X}}) \) when

\[
  OPT(P_1^{\mathcal{X}}) = 3K y^*^2 - \phi, 
\]

where

\[
x_0 = x_0^*, \\
q_i = \frac{1 - x_0^*}{\bar{Q}},
\]

and the CPU-cycle frequency scheduling at the mobile device is

\[
f_w = \frac{B_0 x_0^*}{(1 - x_0^*) \bar{Q}}, \quad \text{for all } w \text{ with } 1 \leq w \leq B_0 x_0^*. 
\]

and the corresponding overall delay is \((1 - x_0^*) \bar{Q})

**Proof.** According to Lemmas 5 and 6 we have

\[
  \min_{x_0, R_{\max}} H(x_0, R_{\max}) = \min \{ h_1(x_0), h_2(x), h_3(x) \}. 
\]

Since \( x_0^* \in [0, \frac{Q}{R + \bar{Q}}] \), we have \( \min h_1(x_0) = h_1(x_0^*) \leq h_1(R^* \bar{Q}) \). As \( h_3(x) \) is a monotonically increasing function, we have \( \min h_3(x) = h_3(R^* \bar{Q}) \).

Notice that \( h_1(x_0) \) and \( h_3(x) \) represent the same point when \( x = R^* \bar{Q} \), thus we have \( h_1(R^* \bar{Q}) = h_3(R^* \bar{Q}) \).
fore, \( \min h_1(x) \leq \min h_3(x) \) holds. Similarly, \( \min h_3(x) \leq \min h_2(x) \). Therefore, we have \( \min h_1(x) \leq \min h_3(x) \leq \min h_2(x) \) and
\[
\min_{x_0, R_{\max}} H(x_0, R_{\max}) = h_1(x_0^*).
\]

Thus, the optimal solution to problem \( P_3 \) is \( h_1(x_0^*) \), which is also the optimal solution to \( P_1 \) as \( P_1 \) to \( P_3 \) is equivalent by Lemma 2. By Lemma 3, \( R_{\max} = (1 - x_0^*)Q \) when \( x_0 \) satisfies Eq. 54. Meanwhile, \( R_{\max} = R_i \) with \( 1 \leq i \leq n \). By the analysis of Case 2. in Section III.C, \( R_{\max} = D_i \) and \( f_w = B_i x_0^*/R_{\max} \). The overall delay \( \max(D_i, R_{\max}) = (1 - x_0^*)Q \).

We now rewrite \( OPT(P_{1}^{\phi}) \) as a function of \( Q \). Eq. 50 can be rewritten as
\[
x_0^* = \frac{y^*}{1 + y^*}.
\]

Hence, \( OPT(P_{1}^{\phi}) \) can be simplified as
\[
OPT(P_{1}^{\phi}) = \frac{K y^* - \phi Q y^*}{Q^2(1 + y^*)} = \frac{K y^* - \phi Q y^*}{Q^2(1 + y^*)} + \alpha Q^3.
\]

Since \( 3K \phi^3 - \phi > 0 \), \( \alpha - 2K \phi^3 > 0 \). Thus, \( \tilde{Q} \) is a strictly monotonically increasing function of \( \phi \). With the property of the inverse function, \( \xi \) is also a strictly monotonically increasing function of \( Q \). Since \( \xi > 0 \), \( OPT(P_{1}^{\phi}) \) is a monotonically increasing function of \( \xi \). Hence, \( OPT(P_{1}^{\phi}) \) is a monotonically increasing function of \( \tilde{Q} \).

We now show that decreasing the value of \( \tilde{Q} \) reduces the overall delay of the execution of task \( A(L, \tau_d) \) by the following lemma.

**Lemma 8.** The overall delay of task \( A(L, \tau_d) \) is a monotonically increasing function with respect to \( \tilde{Q} \).

**Proof.** By replacing \( x_0 \) in Eq. 56 with \( x_0 = 1 - R_{\max}/\tilde{Q} \) by Theorem 1, \( y^* \) can be expressed as follows.
\[
y^* = \frac{\tilde{Q}}{R_{\max}} - 1.
\]

Eq. 51 can be rewritten by replacing \( y^* \) with \( \frac{\tilde{Q}}{R_{\max}} - 1 \) as follows.
\[
2\tilde{Q}^3 - 3\tilde{Q}^2 R_{\max} + \frac{3}{2} R_{\max}^2 = (\frac{\phi}{K} \tilde{Q}^2 + \frac{\alpha}{K} \tilde{Q}^3) R_{\max}^3.
\]

The differentiation of Eq. 61 with respect to \( \tilde{Q} \) then is
\[
K_1 \frac{\partial R_{\max}}{\partial \tilde{Q}} = K_2,
\]

where
\[
K_1 = (\frac{\phi}{K} \tilde{Q}^2 + \frac{\alpha}{K} \tilde{Q}^3 - 1)3R_{\max}^2 + 3\tilde{Q}^2,
\]
\[
K_2 = 6\tilde{Q}^2 - 6\tilde{Q} R_{\max} - 2\frac{\phi}{K} \tilde{Q} R_{\max}^3 - 3\frac{\alpha}{K} \tilde{Q}^2 R_{\max}^3.
\]

The rest is to show that both \( K_1 > 0 \) and \( K_2 > 0 \). As \( R_{\max} = (1 - x^*)Q \), \( Q \geq R_{\max} > 0 \). By Eq. 63, we have
\[
K_1 > -3R_{\max}^2 + 3\tilde{Q}^2 \geq 0.
\]
Meanwhile, by Eq. 61, we have
\[
2\bar{Q}^2 - 3\bar{Q}R_{\text{max}}^3 + \frac{R_{\text{max}}^2}{\bar{Q}} = \left(\frac{\phi}{K} + \frac{\alpha}{K} \bar{Q}^2\right)R_{\text{max}}^3.
\]
(65)

Following Eq. 64 and Eq. 65, the following inequality holds.
\[
K_2 > 6\bar{Q}^2 - 6\bar{Q}R_{\text{max}}^3 - 3\left(\frac{\phi}{K} \bar{Q} R_{\text{max}}^3 + \frac{\alpha}{K} \bar{Q}^2 R_{\text{max}}^3\right)
- 3R_{\text{max}}^3 - \frac{R_{\text{max}}^2}{\bar{Q}}
= 3\bar{Q}R_{\text{max}} - 3R_{\text{max}}^3 - \frac{R_{\text{max}}^2}{\bar{Q}}
= 3\bar{Q} - 3R_{\text{max}}^2 + \frac{R_{\text{max}}}{\bar{Q}}
\geq 0
\]
Having both \( K_1 > 0 \) and \( K_2 > 0 \), by Eq 62, we have \( \frac{\partial R_{\text{max}}}{\partial \bar{Q}} > 0 \).

By Theorem 1, the overall delay of the execution of task \( A(L, \tau_d) \) is equal to \( R_{\text{max}} \), the overall delay thus is a monotonically increasing function of \( \bar{Q} \).

Lemma 7 and Lemma 8 indicate that a smaller \( \bar{Q} \) will result in a less overall delay and cost on the execution of task \( A(L, \tau_d) \). In the following we aim to determine the range of \( \bar{Q} \).

We first determine an upper bound on \( \bar{Q} \) to meet the overall delay requirement of task \( A(L, \tau_d) \). We have \( (2 - \frac{\tau_d}{\phi} \bar{Q}^3 - \frac{(\tau_d + \bar{Q}^2)^2}{\bar{Q}^2} \) is equal to zero by replacing \( R_{\text{max}} \) in Eq. 61 with \( \tau_d \), and let \( \bar{Q}^* \) be the value of \( \bar{Q} \) to ensure the equality holds. By Lemma 8 and the property of the inverse function, \( \bar{Q} \) is a monotonically increasing function of \( R_{\text{max}} \). Since \( R_{\text{max}} \leq \tau_d \), we have \( \bar{Q} \leq \bar{Q}^* \).

We then determine another upper bound on \( \bar{Q} \) to meet the CPU-cycle frequency constraint. By Theorem 1, we have
\[
f_w = \frac{B_0 x_0^*}{Q(1 - x_0^*)} \leq f_{\text{max}}.
\]
(66)

By Eq. 50 and \( \xi = y_m^*/\bar{Q} \) in Lemma 7, Inequality (66) can be rewritten as \( B_0 \xi \leq f_{\text{max}} \). Thus, \( \xi \leq \frac{f_{\text{max}}}{B_0} \).

Eq. 59 can be rewritten by replacing \( \xi \) with \( f_{\text{max}}/B_0 \) as follows.
\[
\bar{Q} = \frac{3K f_{\text{max}}^2 B_0 - \phi B_0^3}{\alpha B_0^2 - 2K f_{\text{max}}^3}.
\]
(67)

Since \( \bar{Q} \) is a monotonically increasing function of \( \xi \) by Lemma 7, \( \bar{Q} \) should be no greater than \( \frac{3K f_{\text{max}}^2 B_0 - \phi B_0^3}{\alpha B_0^2 - 2K f_{\text{max}}^3} \).

Let \( \bar{Q}_{\text{max}} = \frac{3K f_{\text{max}}^2 B_0 - \phi B_0^3}{\alpha B_0^2 - 2K f_{\text{max}}^3} \). Therefore, the range of \( \bar{Q} \) is \( \bar{Q} \leq \min\{\bar{Q}^*, \bar{Q}_{\text{max}}\} \).

Following its definition, the derivative of \( \bar{Q} \) with respect to \( q_i \), where \( 1 \leq i \leq n \), is
\[
\frac{\partial \bar{Q}}{\partial q_i} = \left(\frac{1}{q_i^0}\right) \left(\sum_{i=1}^{n} \frac{1}{q_i}\right)^{-1} > 0.
\]
(68)

Eq. 68 shows that \( \bar{Q} \) is an increasing function of \( q_i \), which implies that a smaller \( q_i \) will correspond a smaller \( \bar{Q} \) for each \( i \) with \( 1 \leq i \leq n \). Thus, to minimize \( \bar{Q} \), we should choose the first smallest \( n \) servers from the \( N \) servers in terms of the value of their \( q_i \).

Let \( j_1, j_2, \cdots, j_N \) be the index sequence of servers by sorting their \( q \)’s in increasing order, i.e., \( q_{j_i} \leq q_{j_{i+1}} \) with \( 1 \leq t \leq N - 1 \). Denote by \( Q(n) \) for any \( n \) with \( 1 \leq n \leq m \leq N \), which is defined as follows.
\[
Q(n) = q_0 + \left(\sum_{i=1}^{n} \frac{1}{q_{j_i}}\right)^{-1}.
\]
(69)

Recall that the number of servers selected to serve task \( A(L, \tau_d) \) cannot exceed \( m \), i.e., \( n \leq m \). Since \( \sum_{i=1}^{n} \frac{1}{q_{j_i}} \leq \sum_{i=1}^{m} \frac{1}{q_{j_i}} \), we have
\[
Q(m) \leq Q(n), \quad 1 \leq n \leq m.
\]
(70)

Function \( Q(n) \) in Inequality (70) reaches the minimum when \( n = m \). Therefore, the number of servers selected to serve task \( A(L, \tau_d) \) should be \( m \) to achieve the minimum overall cost \( OPT(P_1) \) of its execution by Lemma 7. Therefore, we have the following lemma.

**Lemma 9.** Let \( s_0 = \{s_{j_1}, \cdots, s_{j_m}\} \) be the set of servers allocated to task \( A(L, \tau_d) \) in problem \( P_1 \) to achieve the minimum cost \( OPT(P_{1}^{s_0}) \). If \( Q(m) \leq \min\{Q^*, Q_{\text{max}}\} \), then
\[
OPT(P_{1}^{s_0}) = 3K \left(\frac{y_{m}^*}{Q(m)^2}\right) - \phi,
\]
where \( y_{m}^* \) is the value of variable \( y_m^* \) in the following equation.
\[
2y_{m}^* + 3y_{m}^* = \phi, \quad (Q(m)^2 - \phi) \]
(72)

Proof. The subset of servers \( \{s_{j_1}, \cdots, s_{j_m}\} \) can be identified by Ineq. (70). By Theorem 1, Lemma 9 then follows. ■

**B. Algorithm**

Following the discussion in the very beginning in this section, the optimal solution \( OPT(P_0) \) to problem \( P_0 \) is the minimum one between \( OPT_{\text{local}} \) and \( OPT(P_1^{s_{0}}) + E_t + \phi \), while \( OPT_{\text{local}} \) can be expressed as
\[
OPT_{\text{local}} = \min\{\kappa \sum_{w=1}^{B_0} (f_w)^2 + \alpha \sum_{w=1}^{B_0} (f_w)^{-1}\}.
\]
By Eq. 25 and Eq. 27,
\[
OPT_{\text{local}} = B_0(\kappa f^2 + \alpha f^{-1}).
\]
(74)

Therefore, we have
\[
OPT(P_0) = \min\{3K \left(\frac{y_{m}^*}{Q(m)^2}\right) + E_t, \quad B_0(\kappa f^2 + \alpha f^{-1})\}.
\]
(75)

The detailed algorithm for problem \( P_0 \) thus is given in Algorithm 1.

**Theorem 2.** Algorithm 1 for the delay-energy joint optimization problem in an MEC delivers an optimal solution if \( Q(m) \leq \min\{Q^*, Q_{\text{max}}\} \), and its time complexity is \( O(N \log N) \), where \( N \) is the number of servers in the MEC.

Proof. By Theorem 1 and Lemma 9 Algorithm 1 can find an optimal solution to the delay-energy joint optimization problem if \( Q(m) \leq \min\{Q^*, Q_{\text{max}}\} \). Its time complexity analysis is quite obvious, which is \( O(N \log N) \) due to sorting. ■

**V. PERFORMANCE EVALUATION**

In this section, we evaluate the performance of the proposed algorithm for the delay-energy joint optimization problem through simulations. We also investigate the impacts on
Algorithm 1 Task Offloading Scheduling (TOS)

Input: $N$, $m$ and $q_i$ with $1 \leq i \leq N$.
Output: $x_i$ and $f_w$.
1: $x_i \leftarrow 0$, where $i = 0, 1, \cdots, N$;
2: A sequence of servers $j_1, j_2, \cdots, j_N$ is obtained by sorting their $q_i$ in increasing order;
3: Calculate $Q(m)$ and $y_m$ by Eq. (60) and (72), respectively;
4: Calculate $OPT(P_i, x_0)$ by Eq. (71);
5: Calculate $OPT_{local}$ by Eq. (74);
6: Calculate $OPT(f_i)$ by Eq. (75);
7: if $OPT(f_i) = OPT_{local}$ then
8: return $x_0 \leftarrow 1$ and $f_w \leftarrow f$, EXIT.
9: else
10: Calculate $x_i^*$ by Eq. (56);
11: $Q \leftarrow \sum_{i=1}^{m} \frac{1}{q_j}$;
12: Calculate $x_{j_k}$ for all $k$ with $1 \leq k \leq m$ by Eq. (54);
13: Calculate $f_w$ by Eq. (53);
14: return $x_{j_k}$ for all $k$ with $1 \leq k \leq m$, and $f_w$.
15: end if

important parameters on the performance of the proposed algorithm.

A. Experimental settings

We set the number of servers in an MEC at $N = 100$, the data transmission rate between the AP and a server $r_i$ is uniformly distributed between 100 Mbps and 1 Gbps. The computing capacity of a server $c_i$ is uniformly distributed between 1 GHz and 4 GHz. The transmission rate between the mobile device and the AP is set at $r_{h,p} = 2.5$ Mbps and the transmission power of the mobile device $P_x = 0.5$ W. Besides, $E_t = 0.15 J$, $\kappa = 10^{-26}$, $f_{max} = 2$ GHz, and the size of task $A(L, \tau_d)$ is $L = 50$ KB [15]. Assume that the computation intensity of the task is 700 cycles per bit [17]. Under these experimental settings, $Q(m) \leq \min\{Q^*, Q_{max}\}$ holds.

To evaluate the proposed algorithm, Algorithm 1, three benchmark algorithms are introduced as follows.

- **Local_Execution**: The entire task is executed at the mobile device and the CPU-cycle frequencies at the mobile device are scheduled to minimize the overall cost. The optimal value of this policy is $OPT_{local}$.
- **MEC_Execution**: The entire task is offloaded to MEC. Then the subset of servers to serve the task and the task allocation among the selected servers are both optimal. The difference between MEC Execution and TOS is that $x_0 = 0$ in MEC Execution policy.
- **Mixed_Execution**: The task is executed in both the mobile device and MEC. The difference between algorithms Mixed_Execution and TOS is in the setting of $x_0 = 1/(1 + m)$ in algorithm Mixed_Execution.

B. Performance evaluation

We first evaluate the performance of algorithm TOS against benchmark algorithms Local_Execution, MEC_Execution, and Mixed_Execution, by varying both numbers of servers $m$ and $\alpha$. The overall costs delivered by different algorithms are normalized based on the overall cost delivered by TOS.

Fig. 2 shows the overall costs of task execution delivered by the mentioned algorithms. We can see from Fig. 2(a) that the overall cost by algorithm TOS is around 80% of those by algorithms MEC_Execution and Mixed_Execution, and 34% of that by algorithm Local_Execution when $\alpha = 20$. In Fig. 2(b), $m = 5$. Similarly, we can see from Fig. 2(b) that algorithm TOS outperforms the other algorithms.

C. Impact of different parameters

We then study the impacts of important parameters: $m$ and $\alpha$ on the overall delay and energy consumption of the proposed algorithm TOS.

We investigate the impact of the number of chosen servers $m$ on the performance of algorithms TOS, Local_Execution, MEC_Execution, and Mixed_Execution in terms of the overall delay and the energy consumption on the mobile device. We vary $m$ from 1 to 200 while fixing $\alpha$ at 20. It can be seen from Fig. 3(a) that algorithm TOS has the minimum overall delay because of its optimal task allocation among the mobile device and the selected servers. Particularly, when $m \leq 8$, it greatly impacts the overall delay by the mentioned algorithms; otherwise, its impact is negligible.

Fig. 3(b) shows that algorithm TOS is not energy-efficient as it requires local processing at the mobile device to minimize the overall delay when $m \leq 8$; otherwise, its energy consumption impact is negligible.

We also evaluate the impact of parameter $\alpha$ on the performance of the mentioned algorithms in terms of the overall delay and energy consumption. It can be seen from Fig. 4(a) that the increase on the value of $\alpha$ will shorten the overall delay of the task execution when $\alpha \leq 70$; otherwise, its impact on overall delay is negligible. However, as shown in Fig. 4(b), algorithms TOS and Local_Execution will incur more energy consumption at the mobile device.

VI. RELATED WORK

MEC has attracted lots of attention from both industries and academia, see [14] for a comprehensive survey.

Many previous studies on MEC focused on designing a system to support task offloading and allocation [7], [10], [12], [15], [19], [20], [22], [23]. Several offloading frameworks including MAUI [7] and ThinkAir [10], were proposed to prolong the battery lifetime of mobile devices and reduce processing delays of computation tasks. Particularly, Liu et al. [12] introduced a delay-optimal task scheduling policy with random task arrivals for a single-user MEC system. For multi-user MEC systems, Chen et al. [5] proposed a distributed offloading algorithm which can achieve the Nash equilibrium to reduce wireless interferences. You et al. [22] designed a centralized task offloading system for MEC based on time-division multiple access and orthogonal frequency-division multiple access. All of these mentioned work assumed non-adjustable processing capabilities of CPUs at mobile devices. However, it is energy inefficient as the energy consumption of a CPU increases super-linearly with its CPU-cycle frequency [4].

Several recent research has applied the dynamic voltage and frequency scaling (DVFS) technique into task offloading in an MEC environment. For example, Zhang et al. [23] proposed an
energy-optimal binary offloading policy, i.e., either executing a task on a mobile device or offloading the entire task to MEC, under stochastic wireless channels for a single-user MEC system with the DVFS technique. In addition, Sardellitti et al. [20] proposed an algorithm for both communication and computational resource allocations in multi-cell MIMO cloud computing systems. Mao et al. [15] studied an MEC system with energy harvesting devices and focused on reducing the energy consumption of mobile devices by scheduling CPU-cycle frequencies.

VII. CONCLUSION

In this paper, we studied task offloading from mobile devices to MEC. We formulated a novel delay-energy joint optimization
We first formulated the problem as a mixed-integer nonlinear program problem. We then performed a relaxation to relax the original problem to a nonlinear programming problem whose solution can be found in polynomial time, and showed how to make use of the solution to the relaxed problem to find a feasible solution to the problem of concern in this paper. We finally evaluate the performance of the proposed algorithm thorough experimental simulations. Experimental results demonstrated that the proposed algorithm outperforms the three baselines.

APPENDIX

A. Proof of Inequalities \[A\]

Define $\varphi(x) = \frac{1}{x^2}$ with $x > 0$, which is a convex function. Let $\lambda_w = \frac{1}{B}$, where $1 \leq w \leq B$. By Jensen’s Inequality, we have

$$\varphi \left( \sum_{w=1}^{B} \lambda_w x_w \right) = \sum_{w=1}^{B} \lambda_w \varphi(x_w) \geq \varphi \left( \frac{1}{B} \sum_{w=1}^{B} x_w \right) \tag{76}$$

Let $x_w = \frac{1}{f_w}$, where $1 \leq w \leq B$. Recall $R' = \sum_{w=1}^{B} \frac{1}{f_w}$. The left side of Inequality (76) is $\varphi \left( \sum_{w=1}^{B} \lambda_w x_w \right) = \varphi \left( \frac{1}{B} \sum_{w=1}^{B} x_w \right) = B^2 / R'^2$ and the right side is $\sum_{w=1}^{B} \lambda_w \varphi(x_w) = \frac{1}{B} \sum_{w=1}^{B} f_w$. Hence, by Inequality (76) we have $\sum_{w=1}^{B} f_w^2 \geq \frac{B^2}{R'}$. The equality holds when $f_w = B / R'$. Recall $R' \leq R_{\text{max}}$. Inequalities (32) thus holds.