Cloud vs Edge Computing for Mobile Services: Delay-aware Decision Making to Minimize Energy Consumption

Meysam Masoudi, Student Member, IEEE, Cicek Cavdar, Member, IEEE

Abstract

A promising technique to provide mobile applications with high computation resources is to offload the processing task to the cloud. Mobile cloud computing enables mobile devices with limited batteries to run resource hungry applications with the help of abundant processing capabilities of the clouds and to save power. However, it is not always true that cloud computing consumes less energy compared to mobile edge computing. It may take more energy for the mobile device to transmit a file to the cloud than running the task itself at the edge. This paper investigates the power minimization problem for the mobile devices by data offloading in multi-cell multi-user OFDMA mobile cloud computing networks. We consider the maximum acceptable delay and tolerable interference as QoS metrics to be satisfied in our network. We formulate the problem as a mixed integer nonlinear problem which is converted into a convex form using D.C. approximation. To solve the optimization problem, we have proposed centralized and distributed algorithms for joint power allocation and channel assignment together with decision making. Our simulation results illustrate that by utilizing the proposed algorithms, considerable power saving could be achieved e.g. about 60% for short delays and large bitstream sizes in comparison with the baselines.

Index Terms

Offloading, Resource Allocation, Mobile Cloud Computing, Mobile Edge Computing.

I. INTRODUCTION

Swift growth in the development of resource hungry mobile applications has motivated users to use smart phones as a platform for running the applications. However mobile devices cannot

Part of this work has been accepted in IEEE WCNC2017 [1].

This study is supported by EU Celtic Plus Project SooGREEN Service Oriented Optimization of GREEN mobile networks.
always be considered as a platform for resource hungry applications due to their limited power and processing capacity. Moreover, one of the key concerns of users is the battery lifetime of mobile devices [2] while running the applications, knowing the fact that increasing the clock frequency of a CPU increases the power consumption [3]. Therefore, there is a tension between the resource hungry applications and resource poor mobile devices. To tackle the aforementioned problem, one solution is to bridge the gap between available and required resources by offloading the burden from mobile devices to the cloud [4]. Cloud computing with abundant processing resources has become an attractive solution in order to ease this pain for the storage and data processing. Cloud computing for mobile applications will enable new services for mobile users. It is true that cloud computing can potentially save energy for the mobile users [2], however this is not always true when the device consumes more energy to transmit the data to the cloud than to process that data itself [5]. Because of the interference and radio channel conditions, the transmission of the data may consume more energy for the mobile device. However it is not trivial to decide after making a simple comparison of two energy figures for each device served by one base station since the decision may create interference and change the channel conditions for neighboring devices in the surrounding cells. There is also another important factor which has an impact on the decision: delay. A decision making procedure must consider the delay sensitivity of the applications to determine whether to choose local processing or offloading. Mobile devices consume more joules per bit as the delay requirement gets more stringent to process a certain task [6]. Delay requirements from different mobile broadband services can be seen in Table I. In this paper, we investigate the energy saving potential of data offloading in mobile devices under multi-cell multi-user scenario and propose efficient algorithms to make decisions simultaneously for mobile devices to minimize the total energy consumption by meeting the delay requirements from the services. Channel assignment and power allocation problems are considered jointly with the offloading-decision.

A. Related Works

Mobile cloud computing (MCC) provides infrastructure, platform, and software as services to the mobile users [9]. On the other hand, the interaction between cloud and mobile user is inevitable in MCC. Consequently, once users decide to offload data to the cloud, it is necessary to efficiently utilize the available resources. Otherwise, users can not benefit from the potential advantages of MCC. In other words, resource management schemes are the key techniques to
TABLE I: Acceptable delay for different services

| Service Type         | Acceptable Delay [7], [8] |
|----------------------|---------------------------|
| Online Games         | < 1000 ms                 |
| Omnypresent          | 1000 ms                   |
| Third person avatar  | 500 ms                    |
| First person avatar  | 100 ms                    |
| Audio services       | < 450 ms                  |
| Voice over IP        | 200 ms                    |
| Video Services       | < 70 ms                   |
| Video over IP        | 150 ms                    |
| Data                 | < 400 ms                  |
| Non real-time services| Few seconds               |

The conducted surveys in [11] and [12], addressed the existing studies on the integrating mobile edge computing (MEC) to the mobile networks, the computation offloading schemes, resource management problems, and their current challenges. Accordingly, the main focus of [13] is to model the energy consumption of applications in the MCC networks. The authors also proposed an energy aware resource allocation algorithm and scheduling in the cloud. A framework for offloading the computation to the cloud is proposed in [14]. They investigated an offloading infrastructure which eased the migration of the code to the cloud. The main goal of [15] is to study the mobile code offloading architecture. They illustrated that significant energy saving can be obtained by using their offloading methods. Task offloading for different applications for one user case is studied in [16]. In [17], using experimental measurements, it is shown that wireless access has an inevitable effect on the performance of MCC. The authors in [18], considered the problem of resource scheduling for multi-service multi-user MCC networks. Also in [19], a heuristic approach is adopted to minimize the energy consumption of all users while making decision on offloading and resource allocation for each task. The authors in [20], modeled the energy consumption of the mobile devices. They formulated an optimization problem to minimize the energy consumption of a single device by data offloading. A dynamic application’s task offloading algorithm using Lyapunov optimization is proposed in [21], aiming at minimizing the energy...
The authors in [22], presented a practical offloading framework in a cost aware Wi-Fi system considering the throughput-delay trade offs. In [23], a game theoretic approach is adopted to design an offloading mechanism for mobile devices. In their model, a multi-user case has been considered while the corresponding QoS as well as their effect on the other users are ignored. In [24], a decentralized offloading game is proposed to make decision among mobile devices in a simple single channel scenario. The partitioning problem for mobile data stream application is defined in [25]. They have used genetic algorithm to solve the problem. They also reported that partitioning data can enhance the application performance in terms of throughput. The authors in [26], utilized the Markov decision process approach to solve the problem of task offloading. They have formulated a delay minimization problem to find the optimal task scheduling policy. In [27], the authors studied the problem of network energy minimization in C-RAN based, MCC system. In this study, the authors jointly optimized the beamforming design and power allocation with a decision making strategy. For energy consumption and latency minimization problem, partial computation offloading algorithm to optimize the computational speed of mobile devices and their transmit power is proposed in [28]. In [29], the authors deal with the latency issue by means of cloudlet infrastructure, which is a data center to bring the cloud closer to the users. The authors in [6], proposed a model for the mobile device energy consumption. They have also derived an offloading policy considering delay and energy consumption under single stochastic wireless channel with only ”good” or ”bad” channel state. Their model is limited to single-user single-channel case and interference and users’ QoS is not addressed in their model. The authors in [30], considered a simple single-user mobile-edge computing system. They proposed an algorithm to optimize the power consumption and to minimize the delay. In this study, the interference analysis and its effect on the offloading decision is missing. The authors in [31], solved the offloading optimization problem to remove the processing burden from mobile devices without considering the resource allocation. The authors in [32], modeled the offloading decision as a competitive game where users try to minimize their energy consumptions. They did not consider the power allocation which has significant impact on the performance of the algorithms. In [33], to minimize the offloading energy consumption, the authors proposed the joint optimization of computing and radio resources considering the latency constraints in a cloud-edge computing network. In [1], we proposed joint power allocation, decision making and channel assignment (J-PAD) algorithm to perform the resource allocation considering interference
and delay constraints.

B. Contributions

There are still plenty of challenges to be tackled in the multi-cell multi-user and multi-channel MCC networks. To the best of our knowledge, the problem of resource allocation and decision making for data offloading in multi cell networks considering multi users has not been addressed in the literature. In this paper we aim at minimizing the power consumption of users while considering the user’s QoS in terms of delay and maximum tolerable interference on each channel. We formulate the resource allocation and offloading optimization problem. We show that the problem is mixed integer nonlinear problem (MINLP), where the optimal solution is intractable. To have a tractable solution, we convert the problem to the convex form and propose two algorithms called J-PAD and C-PAD to solve the problem in a polynomial time. The main contribution of this paper can be summarized as follows:

- In the context of multi cellular multi user OFDMA MCC networks, we formulate the resource allocation and offloading problem that is aware of network status and users’ demand aiming at minimizing the total power consumption of all users subject to constraints on QoS of users and interference threshold.
- We formulated the problem as a mixed integer nonlinear optimization problem (MINLP). To solve the problem, it is converted to the convex form using variable changing, DC approximation, adding penalty factor, and relaxing the binary constraints. Therefore the problem can be solved in a polynomial time.
- We also propose two algorithms to solve the problem of resource allocation and decision making. The first algorithm is a centralized scheme, designed to be performed at the base station while the second one is a distributed scheme, which requires a partial information exchange, suitable to be performed at the user terminal. The complexity of these algorithms is also investigated.
- Through simulations, we show that there exists an offloading region for each user where offloading can help them to save more power. By comparing the cell edge user and normal user in the network, we show that the optimal region depends not only on delay threshold and bit stream size of users but also on the position and channel condition of the users.

The rest of the paper is organized as follows. In Section II system model is presented. The problem formulation and the solution methodology are discussed in Section II-C. We propose
our algorithms and corresponding complexity analysis in Section IV followed by the simulation results presented in Section V. Finally, we bring the concluding remarks in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Description

According to Fig. 1, we consider a cellular network with $N_c$ base stations where mobile users (MUs) are uniformly distributed within a cell range. Each base station is equipped with a server which is responsible for the offloaded users’ data processing and we assume there is a centralized unit which exchanges the required information between base stations using backhaul. Each cell can serve up to $F_i$ active users. We assume that the available bandwidth $B$ is divided to $N$ sub-channels. The sub-channel model is adopted from [34] and is composed of large scale fading, small scale fading, and shadow fading. Also, we consider OFDMA as an access method, hence users in the same cell cannot share same sub-channel with each other; however, each user might experience an interference from neighboring cells. In this model, user $j$ in cell $i$ has a bit stream.
of size $L_{i,j}$. We have generated the users’ bit stream size with normal distribution with mean $L_{i,j}$ and variance $\frac{1}{10}L_{i,j}$. Users can process the data on their own or send it to the cloud. Users can not use both schemes, e.g. sending a portion of the data to the cloud and processing the remaining data locally. The data corresponds to the user $j$ in cell $i$ should be processed within the maximum acceptable delay (delay threshold), $T_{i,j}$, generated with normal distribution with mean $T_{i,j}$ and variance $\frac{1}{10}T_{i,j}$. As we assume that the processed data is short, the response time delay can be neglected [6].

B. Power Model

1) Local Processing Power Model: When users are supposed to process the data locally, the CPU power consumption is dominant. It is composed of dynamic power, circuit power, and leakage power [28]. The dynamic power as a dominating power in CPU, is function of required CPU cycles which depends on both delay threshold and input data size. Under the optimal value for CPU frequency, the minimum power consumption of CPU is proportional to the $(T/L)^n$, where $T$ is the maximum acceptable delay and $L$ is the users’ bit stream size and $n$ is the scaling factor power [6]. Consequently, we use the following model for local processing power consumption:

$$p_{\text{Local}}^{i,j} = M \frac{L^n_{i,j}}{T^n_{i,j}},$$

(1)

where $p_{\text{Local}}^{i,j}$ is the local processing power consumption of user $j$ in cell $i$ and $M$ is a constant value that depends on the CPU and application parameters [6].

2) Offloading Power Model: The transmission power, for sending data to the cloud is,

$$p_{\text{tx}}^{i,j} = \sum_{n=1}^{N} a_{i,j,n} p_{i,j,n},$$

(2)

where $p_{\text{tx}}^{i,j}$ denotes the transmission power consumption of user $i$ in cell $j$ and $a_{i,j,n}$ is a binary variable representing whether the corresponding sub-channel is assigned to the user or not. Therefore, the user’s total transmission power is

$$P_{\text{tx}}^{i,j} = \frac{1}{\eta} p_{\text{tx}}^{i,j} + p_c,$$

(3)

where $\eta$ is power amplifier coefficient and $p_c$ is a constant circuit power.
3) Aggregated Power Model: Total power consumption of the active users in the network can be written as:

\[ P_{\text{Total}} = \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} p_{i,j} \]  

(4)

where

\[ p_{i,j} = s_{i,j} p_{i,j}^{\text{Tx}} + (1 - s_{i,j}) p_{i,j}^{\text{Local}} \]

\[ = s_{i,j} \left( \frac{1}{\eta} \sum_{n=1}^{N} a_{i,j,n} p_{i,j,n} + p_c \right) \]

\[ + (1 - s_{i,j}) \frac{ML_{i,j}^n}{T_{i,j}^n}. \]

(5)

The integer variable \( s_{i,j} \) takes the value of 0 if user \( j \) in cell \( i \) uses its own processor and takes the value of 1 if the user sends the data to the cloud. Therefore, the total power consumption can be written as:

\[ P_{\text{Total}} = \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} s_{i,j} \left( \frac{1}{\eta} \sum_{n=1}^{N} a_{i,j,n} p_{i,j,n} + p_c \right) \]

\[ + \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} (1 - s_{i,j}) \frac{ML_{i,j}^n}{T_{i,j}^n}. \]

(6)

Moreover, the signal to noise plus interference ratio at the base station in cell \( i \) is given by:

\[ \gamma_{i,j,n} = \frac{p_{i,j,n} h_{i,j,n}}{\sigma^2 + I_{i,j,n}^{(n)}}, \]

(7)

where the channel gain from \( j \)-th MU of \( i \)-th cell is denoted by \( h_{i,j,n} \). The channel gain from user \( m \), in cell \( k \) to the cell \( i \) is denoted by \( h_{k,m,n}^i \). The first term in the denominator of (7) is the noise power and the second term is the interference from other cells on channel \( n \) in cell \( i \) which can be calculated as:

\[ I_{i,j,n}^{(n)} = \sum_{k=1}^{N_c} \sum_{m=1}^{F_k} s_{k,m} p_{k,m,n}^{tx} h_{k,m,n}^i. \]

(8)

In our assumption, the users must utilize the whole duration. Considering fixed \( T \), power minimization is in line with energy minimization.
C. Problem Formulation

In this section, we develop the mathematical formulation for decision making and resource allocation problem. The base station determines the offloading users and allocates sub-channels to its users and specifies the suitable power level on each sub-channel. The objective of the resource allocation is to minimize the aggregated power consumption of all users by allocating resources to the offloading users in a way that their delay requirement is satisfied. The optimization problem can be formulated as follows:

\[
\begin{align*}
\min_{\{a, p, s\}} & \quad \mathbf{P}_{Total} \\
\text{subject to} & \\
C1: & \quad 0 \leq p_{i,j}^{Tx} \leq p_{max}, \quad \forall i, j, \\
C2: & \quad \sum_{k=1, k\neq i}^{N_c} \sum_{j=1}^{F_i} s_{k,j} a_{k,j,n} p_{k,j,n} h_{k,j,n}^{i} \leq I_{th}^{(n)}, \quad \forall i, n, \\
C3: & \quad T_{i,j} \leq T_{max}, \quad \forall i, j, \\
C4: & \quad \sum_{j=1}^{F_i} \sum_{n=1}^{N} s_{i,j} a_{i,j,n} \log_2(1 + \gamma_{i,j,n}) \leq R_{Proc}^{max}, \quad \forall i, \\
C5: & \quad \sum_{j=1}^{F_i} a_{i,j,n} \leq 1, \quad \forall i, n, \\
C6: & \quad a_{i,j,n} \in \{0, 1\}, \quad \forall i, j, n, \\
C7: & \quad s_{i,j} \in \{0, 1\}, \quad \forall i, j.
\end{align*}
\]

In (9), the objective is to minimize the total power consumption of all active MUs in the network. The constraint C1 indicates that the transmit power of each user is limited to \( p_{max} \). The constraint C2 states that for each base station \( i \), the interference arising from other cells on each sub-channel is restricted to be within a threshold. The constraint C3 restricts the maximum tolerable delay for user \( j \) in the \( i \)-th cell to \( T_{max} \) if the aforementioned user sends its data to the cloud. If a user decides to process the data locally, then the CPU will be responsible for satisfying this constraint. In our analysis we assume that CPU uses the entire available time to reduce the power consumption. The constraint C4 addresses the processing limitation at the cloud. The constraint C5 guarantees the OFDMA assumption in each cell where each sub-channel
is assigned to at most one user. The constraints C6 and C7 indicate that the sub-channel and data offloading indices are binary variables. It is worth mentioning that the constraint C3 can be written in an equivalent form. Using C3 we will have

$$\frac{L_{i,j}}{T_{i,j}} \geq \frac{L_{i,j}}{T_{max}}.$$  \hspace{1cm} (10)

Defining $R_{min} \triangleq \frac{L_{i,j}}{T_{max}}$ and noting that the left side of (10) is the total data rate of the $j$-th user in the $i$-th cell, we obtain the following equivalent constraint for C3:

$$s_{i,j} \sum_{n=1}^{N} a_{i,j,n} \log_2(1 + \gamma_{i,j,n}) \geq s_{i,j} R_{min}, \forall i, j.$$  \hspace{1cm} (11)

In the rest of this paper, we consider the constraint C3 in the form presented in (11).

The optimization problem defined in (9) is a mixed integer nonlinear problem (MINLP) and finding the optimal solution is NP-hard and cannot be solved in a polynomial time. The non-convexity is coming from three reasons in the problem. The first and second reasons are the binary inherent of decision making variable, constraint C7, and the combinatorial nature of sub-channel allocation, constraint C6. The third one is due to the constraints C3 and C4 and existence of the power allocation variable in the denominator of SINR formula defined in (7). In the following section, we address how to deal with these variables and solve the problem by converting it into a convex form.

### III. Solution Methodology

In this section, we aim to transform the primary problem defined in Section II-C into a canonical convex form. In this regard, we classify the challenges into two categories, binary variables and non convex functions.

To resolve the challenges caused by the binary variables, one approach is to relax the troublesome constraints, sub-channel allocation for instance, to shape the problem into a convex form and then making hard decision in the end as we did in [35]. An alternative approach is to add auxiliary constraints to enforce the solution to be in our desired form as we will describe later. Another approach is to break the problem into sub-problems so that one could successively first solve the problem for the annoying binary variable and consequently, given this variable, solve the rest of the problem.

To deal with the non-convex functions, we utilize a theory of optimization for a superclass of convex functions, called Difference of Convex (D.C.) functions [36]. Later we demonstrate that
our problem can be written in form of D.C. functions. In the end, applying Taylor approximation enables us to solve the last stage of converting the primary problem defined in (9), into a convex form. Having all these powerful approaches available, we tackle the problem, as follows.

In the first step, we break down the problem into two sub-problems and then solve them successively. The first sub-problem is to determine the channel assignment for each user in each cell. The second sub-problem is to find out the decision variable and power allocation. We use the solution of the first sub-problem as an input to the second sub-problem. Also, the results of second sub-problem is used to update the solution for the first sub-problem and this process continues until the convergence. Furthermore, we apply two approaches to solve the second sub-problem. The overview of two utilized approaches to solve the problem can be seen in equations (12) and (13).

In the first approach, after separating the sub-channel assignment, the problem can be solved jointly for other variables e.g. power allocation and decision variable as follows:

\[
\begin{align*}
\text{Initialization} & \quad a[0] \rightarrow (p[0], s[0]) \rightarrow \ldots \rightarrow a[t-1] \rightarrow (p[t-1], s[t-1]) \\
\text{Iteration } t-1 & \quad \rightarrow a[t] \rightarrow (p[t], s[t]) \rightarrow a^* \rightarrow (p^*, s^*). 
\end{align*}
\]

(12)

In the second approach, we separate sub-channel assignment, power allocation, and decision variable from each other as follows:

\[
\begin{align*}
\text{Initialization} & \quad a[0] \rightarrow \tilde{p}[0] \rightarrow s[0] \rightarrow \ldots \rightarrow a[t-1] \rightarrow \tilde{p}[t-1] \rightarrow s[t-1] \\
\text{Iteration } t & \quad \rightarrow a[t] \rightarrow \tilde{p}[t] \rightarrow s[t] \rightarrow a^* \rightarrow \tilde{p}^* \rightarrow s^*. 
\end{align*}
\]

(13)

The main difference between these two approaches is that in the former, we jointly solve the problem of power allocation and decision making; However, in the latter, we divide the second sub-problem into two steps and solve each sub-problem individually. In the following subsections, first we deal with solving the first sub-problem followed by solving the second sub-problem by converting it into a convex form.

A. Sub-Problem One: Optimal Sub-channel Assignment

Given the power allocation vector \(p[t-1]\), the optimal sub-channel assignment \(a[t]\) for further power allocation and offloading in the next iteration \(t\) is as follows:
**Proposition 1.** Given the power vector, minimum power consumption is attained when each sub-channel in each cell is assigned to the MU with the highest effective interference on that sub-channel.

**Proof.** Because the problem is power minimization and also minimum data rate requirement of users should be satisfied, the minimum power is consumed when the inequality of minimum required rate becomes the equality. Now let us assume that all users are given the best possible channel to reach their data rate with minimum power consumption. Also, let a user have a channel with effective interference value lower than a highest value and the user has data rate $r_{\text{min}}$ on that channel. Thus, the consumed power on that channel is

$$\log_2(1 + \gamma_{i,j,n}) = r_{\text{min}}$$

$$p_{i,j,n} = \frac{C}{\frac{h_{i,j,n}}{\sigma^2 + I_n}}$$

where $C$ is a constant value. Also from our assumption, we know that the effective interference in a denominator of (15), e.g. $\frac{h_{i,j,n}}{\sigma^2 + I_n}$, is not the highest possible value. Hence, if we assign the highest effective interference value to this user, the total power consumption will be lower and this is in contrast with the assumption of minimum power consumption. Therefore, minimum power is consumed when maximum effective interference is the criterion for the channel allocation. In other words, with higher effective interference, less power is consumed to satisfy the minimum required rate.

Let $EI_{i,j,n}$ denotes the effective interference vector of a user on the channel $n$. High effective interference in a channel means that the MU is experiencing a good channel condition with a low interference from other cells. Therefore, the decision for channel allocation will be made based on the following criterion:

$$\tilde{a}_{i,j,n} = 1 \bigg| n = \max EI_{i,j,n} \forall i, j.$$  

Thus, a channel allocation matrix $\tilde{a}[t]$ at time $t$, can be formed with the elements obtained from the equation (16).

At this stage we have solved the first sub-problem and the results will be available for next steps. In the next two subsections, we solve the second sub-problem introduced in (12) and (13).
B. Sub-Problem Two: Power Allocation, and Decision Making

In the previous subsection we have solved the problem of sub-channel assignment and therefore one of the challenges of the primary problem (9) is resolved. The results of previous subsection will be used in this section to solve the sub-problem of power allocation and decision making. As in (12) and (13), two approaches are applied to tackle the challenges. These approaches are discussed in the following subsections.

1) Joint Power Allocation and Decision Making (J-PAD): Given a sub-channel assignment, the problem of joint power allocation and data offloading can be rewritten as:

\[
\begin{align*}
\min_{\{p, s\}} & \quad P_{\text{Total}} \\
\text{subject to} & \\
C1: & \quad 0 \leq s_{i,j}P_{i,j}^{\text{Tx}} \leq p_{\text{max}}, \quad \forall i, j, \\
C2: & \quad \sum_{k=1}^{N_c} \sum_{j\neq i} s_{k,j}p_{k,j,n}h_{k,j,n}^{i} \leq I_{\text{th}}^{(n)}, \quad \forall i, n, \\
C3: & \quad s_{i,j} \sum_{n=1}^{N} \log_2(1 + \gamma_{i,j,n}) \geq s_{i,j}R_{\text{min}}, \quad \forall i, j, \\
C4: & \quad \sum_{j=1}^{F_i} \sum_{n=1}^{N} s_{i,j} \log_2(1 + \gamma_{i,j,n}) \leq R_{\text{Proc}}^{P_{\text{max}}}, \quad \forall i, \\
C7: & \quad s_{i,j} \in \{0, 1\}, \quad \forall i, j.
\end{align*}
\]

To solve (17), we first reformulate it to a more mathematically tractable form. Since $s_{i,j}$ is a binary variable, we can write $s_{i,j} \log_2(1 + \gamma_{i,j,n}) = \log_2(1 + s_{i,j} \gamma_{i,j,n})$. Moreover, the problem consists of the product terms of $s_{i,j}P_{i,j,n}$. We use the following change of variable

\[
\tilde{p}_{i,j,n} = s_{i,j}p_{i,j,n},
\]

(18)
to recast the optimization problem. Also, the optimization problem includes integer variable $s_{i,j}$. Hence to convert $s_{i,j}$s into continuous variables, we can express the constraint C7 as the intersection of the following regions:

\[
\begin{align*}
\mathcal{R}_1: & \quad 0 \leq s_{i,j} \leq 1, \forall j, i, \\
\mathcal{R}_2: & \quad \sum_{j} \sum_{i} (s_{i,j} - s_{i,j}^2) \leq 0.
\end{align*}
\]

(19)
Hence, we can write the optimization problem of (17) as follows

$$\begin{align*}
\min_{\tilde{p}, s} \quad & \mathbf{p}^{Total} \\
\text{s.t.} \quad & C1–C4, \mathcal{R}_1, \mathcal{R}_2.
\end{align*}$$

The problem of (20) is a continuous optimization problem with respect to all variables. However, we aim to find integer solutions for $s_{i,j}$’s. To attain this goal, we add a penalty function to the objective function of (20) to penalize it if the values of $s_{i,j}$’s are not integer. Thus, the problem can be modified to

$$\begin{align*}
\min_{\tilde{p}, s} \quad & \mathcal{L}(\tilde{p}, s, \lambda) \\
\text{s.t.} \quad & C1–C4, \mathcal{R}_1.
\end{align*}$$

In (21), $\mathcal{L}(\tilde{p}, s, \lambda)$ is the Lagrangian of (20), and is defined as

$$\mathcal{L}(\tilde{p}, s, \lambda) \triangleq \mathbf{p}^{Total} + \lambda \sum_j \sum_i (s_{i,j} - s_{i,j}^2),$$

where $\lambda$ is the penalty factor which should be $\lambda \gg 1$. It can be shown that, for sufficiently large values of $\lambda$, the optimization problem of (21) is equivalent to (20) and attains the same optimal value [37].

**Proposition 2.** For sufficiently large values of $\lambda$, the optimization problem of (21) is equivalent to (20).

*Proof.* We start with this point that the optimization problem of (21) can be expressed as $\min_{\tilde{p}, s} \max_{\lambda} \mathcal{L}(\tilde{p}, s, \lambda)$ and its dual problem can be written as $\max_{\lambda} \min_{\tilde{p}, s} \mathcal{L}(\tilde{p}, s, \lambda)$. Suppose that $\tilde{p}_\lambda$, $s_\lambda$, and $\varphi(\lambda)$ denote the optimal solution and the optimal value of of the optimization problem of (21), respectively, i.e.

$$\varphi(\lambda) = \mathcal{L}(\tilde{p}_\lambda, s_\lambda, \lambda) = \min_{\tilde{p}, s} \mathcal{L}(\tilde{p}, s, \lambda)$$

Then, we will have

$$\max_{\lambda} \varphi(\lambda) = \max_{\lambda} \mathcal{L}(\tilde{p}_\lambda, s_\lambda, \lambda) = \max_{\tilde{p}, s} \mathcal{L}(\tilde{p}, s, \lambda)$$

$$\leq \min_{\tilde{p}, s} \max_{\lambda} \mathcal{L}(\tilde{p}, s, \lambda) = \text{problem}(20)$$

(24)
Recall that for $\forall s \in D, R_1$, we have

$$\sum_i \sum_j (s_{i,j} - s^2_{i,j}) \geq 0.$$ 

In other words, $\varphi(\lambda)$ is an increasing function in $\lambda$ and according to (24), is bounded by the optimal value of problem (20). If for some $0 \leq \lambda^* < \infty$, $\sum_i \sum_j (s_{i,j} - s^2_{i,j}) = 0$, then $(\tilde{p}_{\lambda^*}, s_{\lambda^*})$ is feasible for the main problem, too. As a result, we will have

$$\varphi(\lambda^*) = \mathcal{L}(\tilde{p}_{\lambda^*}, s_{\lambda^*}, \lambda^*) = \max_{\lambda} \mathcal{L}(\tilde{p}_{\lambda^*}, s_{\lambda^*}, \lambda)$$

$$\geq \min_{\tilde{p}, s} \max_{\lambda} \mathcal{L}(\tilde{p}, s, \lambda)$$

(25)

Comparing (25) and (24), we conclude that the strong duality holds and we have

$$\varphi(\lambda^*) = \max_{\lambda} \varphi(\lambda),$$

(26)

since $\varphi(\lambda)$ is a monotonically increasing function with respect to $\lambda$, for $\forall \lambda \geq \lambda^*$ we have

$$\varphi(\lambda) = p^{Total}(\tilde{P}_{\lambda^*}, s_{\lambda^*}) = \min_{\tilde{p}, s} \max_{\lambda} \mathcal{L}(\tilde{p}, s, \lambda) = \text{problem}(20)$$

(27)

At the optimal point and for the second case where we have $\sum_i \sum_j (s_{i,j} - s^2_{i,j}) > 0$, $\varphi(\lambda^*)$ goes to $\infty$ because of the monotonicity of the $\varphi(\lambda)$ with respect to the $\lambda$. This contradicts the max-min inequality which states that $\varphi(\lambda^*)$ is bounded from above. Thus, the term $\sum_i \sum_j (s_{i,j} - s^2_{i,j})$ should be zero, and the results of the first case hold. $\square$
Now, the optimization problem can be converted to the following problem

\[
\begin{align*}
\min_{\{\tilde{p}, s\}} & \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} \sum_{n=1}^{N} \tilde{p}_{i,j,n} + \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} (1 - s_{i,j}) \frac{M \tilde{l}_{i,j}^3}{T_{i,j}^3} \\
& + \lambda \left( \sum_i \sum_j (s_{i,j} - s_{i,j}^2) \right) \\
\text{subject to} & \\
C1: & 0 \leq \sum_{n=1}^{N} \tilde{p}_{i,j,n} \leq s_{i,j} p_{max}, \quad \forall i, j, \\
C2: & \sum_{k=1 \atop k \neq i}^{N_c} \sum_{j=1}^{F_i} \tilde{p}_{i,j,n} h_{k,j,n}^i \leq I_{th}^{(n)}, \quad \forall i, n, \\
C3: & \sum_{n=1}^{N} \log_2(1 + \frac{\tilde{p}_{i,j,n} h_{i,j,n}^i}{\sigma^2 + I_{th}^{(n)}}) \geq s_{i,j} R_{min}, \quad \forall i, j, \\
C4: & \sum_{j=1}^{F_i} \sum_{n=1}^{N} \log_2(1 + \frac{\tilde{p}_{i,j,n} h_{i,j,n}^i}{\sigma^2 + I_{th}^{(n)}}) \leq R_{Proc}^{max}, \quad \forall i, \\
C7: & s_{i,j} \in [0, 1], \quad \forall i, j,
\end{align*}
\]

where \( \tilde{i}_{i}^{(n)} \triangleq \sum_{k=1 \atop k \neq i}^{N_c} \sum_{m=1}^{F_i} \tilde{p}_{k,m,n} h_{k,m,n}^i \). We can write the objective function in (28) as \( f_1(\tilde{p}, s) - f_2(\tilde{p}, s) \), where \( f_1(\tilde{p}, s) \triangleq \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} \sum_{n=1}^{N} \tilde{p}_{i,j,n} + \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} (1 - s_{i,j}) \frac{M \tilde{l}_{i,j}^3}{T_{i,j}^3} + \lambda s_{i,j} \), and \( f_2(\tilde{p}, s) \triangleq \lambda \sum_{i=1}^{N_c} \sum_{j=1}^{F_i} s_{i,j}^2 \) are two convex functions. In a similar way, for \( \forall i, j \), we define \( z_{i,j,n}(\tilde{p}) \) and \( q_{i,j,n}(\tilde{p}) \) as

\[
\begin{align*}
z_{i,j,n}(\tilde{p}) & \triangleq \log_2 \left( \tilde{p}_{i,j,n} h_{i,j,n} + \sum_{k=1 \atop k \neq i}^{N_c} \sum_{m=1}^{F_i} \tilde{p}_{k,m,n} h_{k,m,n}^i + \sigma^2 \right), \\
q_{i,j,n}(\tilde{p}) & \triangleq \log_2 \left( \sum_{k=1 \atop k \neq i}^{N_c} \sum_{m=1}^{F_i} \tilde{p}_{k,m,n} h_{k,m,n}^i + \sigma^2 \right),
\end{align*}
\]

then, we can write constraints C3 and C4 as follows

\[
\begin{align*}
C3: & \quad Z_{i,j}(\tilde{p}) - Q_{i,j}(\tilde{p}) \geq s_{i,j} R_{min}, \forall i, j, \\
C4: & \quad Q_{i}(\tilde{p}) - Z_{i}(\tilde{p}) \geq -R_{Proc}^{max}, \forall i,
\end{align*}
\]
where \( Z_{i,j}(\tilde{p}) \triangleq \sum_{n=1}^{N} z_{i,j,n}(\tilde{p}) \), \( Q_{i,j}(\tilde{p}) \triangleq \sum_{n=1}^{N} q_{i,j,n}(\tilde{p}) \), \( Z_i(\tilde{p}) \triangleq \sum_{j=1}^{F_i} \sum_{n=1}^{N} z_{i,j,n}(\tilde{p}) \), and \( Q_i(\tilde{p}) \triangleq \sum_{j=1}^{F_i} \sum_{n=1}^{N} q_{i,j,n}(\tilde{p}) \) are concave functions. Therefore, the problem is in the form of the difference of two convex (concave) functions (D.C. programming) [36]. In D.C. programming, we start from a feasible initial point and iteratively solve the optimization problem. Let \( k \) denote the iteration number. At the \( k \)-th iteration, to make the problem convex, using the first order Taylor approximation for \( f_2(\tilde{p}, s) \), \( Q_{i,j}(\tilde{p}) \) and \( Z_i(\tilde{p}) \) as follows

\[
\tilde{f}_2(\tilde{p}, s) \approx f_2(\tilde{p}, s^{k-1}) + \nabla_s f_2(\tilde{p}, s^{k-1}).(s - s^{k-1}),
\]

\[
\tilde{Q}_{i,j}(\tilde{p}) \approx Q_{i,j}(\tilde{p}^{k-1}) + \nabla_{\tilde{p}} Q_{i,j}(\tilde{p}^{k-1}).(\tilde{p} - \tilde{p}^{k-1}),
\]

\[
\tilde{Z}_i(\tilde{p}) \approx Z_i(\tilde{p}^{k-1}) + \nabla_{\tilde{p}} Z_i(\tilde{p}^{k-1}).(\tilde{p} - \tilde{p}^{k-1}),
\]

where \( \tilde{p}^{k-1} \) and \( s^{k-1} \) are the solutions of the problem at \( (k-1) \)-th iteration and \( \nabla_x \) denotes the gradient operation with respect to \( x \). Thus, at the \( k \)-th iteration, instead of dealing with the problem of (17), we solve the following convex problem

\[
\min_{\{\tilde{p}, s\}} f_1(\tilde{p}, s) - \tilde{f}_2(\tilde{p}, s)
\]

subject to: C1, C2, C7,

C3: \( Z_{i,j}(\tilde{p}) - \tilde{Q}_{i,j}(\tilde{p}) \geq s_{i,j}R_{\text{min}}, \quad \forall i,j \),

C4: \( Q_i(\tilde{p}) - \tilde{Z}_i(\tilde{p}) \geq -R_{\text{Proc max}}, \quad \forall i \).

It can be shown that the D.C. programming results in a sequence of feasible solutions that iteratively achieves better solutions than previous iteration until it converges.

**Proposition 3.** The D.C. programming results in a sequence of feasible solutions that iteratively decrease the total power consumption of the network.

**Proof.** To show that our solutions are feasible for the original problem, first, we notice that the solution of the approximated problem in the \( i \)-th iteration must satisfy the constraint C3 and C4, i.e.,

\[
Z_{i,j}(\tilde{p}^i) - \tilde{Q}_{i,j}(\tilde{p}^i) = Z_{i,j}(\tilde{p}^i) - \{Q_{i,j}(\tilde{p}^{i-1}) + \nabla_{\tilde{p}} Q_{i,j}(\tilde{p}^i).(\tilde{p}^i - \tilde{p}^{i-1})\}
\geq R_{\text{min}},
\]

(34)
\[ Q_i(\tilde{p}^t) - Z_i(\tilde{p}^t) = \]
\[ Q_i(\tilde{p}^t) - \{ Z_i(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Z_i^T(\tilde{p}^t) . (\tilde{p}^t - \tilde{p}^{t-1}) \} \]
\[ \geq -R_{Proc}^m, \] (35)

On the other hand, since \( Z_{i,j} \) and \( Q_i \) are two concave functions with respect to \( \tilde{p} \), due to the first order condition for the concave functions [38], we have
\[ Z_{i,j}(\tilde{p}) \leq Z_{i,j}(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Z_{i,j}^T(\tilde{p}^{t-1}) . (\tilde{p}^{t-1} - \tilde{p}^{t-1}). \] (36)
\[ Q_i(\tilde{p}) \leq Q_i(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Q_i^T(\tilde{p}^{t-1}) . (\tilde{p}^{t-1} - \tilde{p}^{t-1}). \] (37)

Substituting \( \tilde{p} = \tilde{p}^t \) into (36) and (37) results in
\[ Z_{i,j}(\tilde{p}^t) \leq Z_{i,j}(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Z_{i,j}^T(\tilde{p}^{t-1}) . (\tilde{p}^{t-1} - \tilde{p}^{t-1}). \] (38)
\[ Q_i(\tilde{p}^t) \leq Q_i(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Q_i^T(\tilde{p}^{t-1}) . (\tilde{p}^{t-1} - \tilde{p}^{t-1}). \] (39)

From (34) and (35), we conclude that
\[ Z_{i,j}(\tilde{p}^t) - Q_{i,j}(\tilde{p}^t) \geq \]
\[ Z_{i,j}(\tilde{p}^t) - \{ Q_{i,j}(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Q_{i,j}^T(\tilde{p}^t) . (\tilde{p}^t - \tilde{p}^{t-1}) \} \]
\[ \geq R_{min}, \] (40)
\[ Q_i(\tilde{p}^t) - Z_i(\tilde{p}^t) \geq \]
\[ Q_i(\tilde{p}^t) - \{ Z_i(\tilde{p}^{t-1}) + \nabla_{\tilde{p}} Z_i^T(\tilde{p}^t) . (\tilde{p}^t - \tilde{p}^{t-1}) \} \]
\[ \geq -R_{Proc}^m, \] (41)

Thus, the solution for the approximated problem is feasible for the original problem too. Now, we show that the total power consumption of the network will decrease iteratively. Since \( g(s) \) is a convex function, due to the first order condition for the convex functions [38], we have
\[ g(s) \geq g(s^0) + \nabla_s g^T(s^{t-1}) . (s - s^0). \] (42)
Using (42) and considering the fact that the objective function of (28) can be written as \( f(s) - g(s) \), at the \((t + 1)\text{-th}\) iteration we have

\[
\begin{align*}
& f(s^{t+1}) - g(s^{t+1}) \leq f(s^t) - g(s^t) - \nabla_s g^T(s^t). (s^{t+1} - s^t) \\
& = \min_s f(s) - g(s^t) - \nabla_s g^T(s^t). (s - s^t) \\
& \leq f(s^t) - g(s^t) - \nabla_s g^T(s^t). (s^t - s^t) = f(s^t) - g(s^t)
\end{align*}
\]

Thus, the total power consumption of the network decreases as iterations continue. \(\square\)

2) Channel assignment, Power Allocation, and Decision Making (C-PAD): Similar to subsection III-B1 we assume that channel assignment vector is given based on proposition 1. Given sub-channel assignment, the optimization problem can be rewritten as:

\[
\min_{\{p\}} P_{Total}^{C-PAD} \tag{43}
\]

subject to

\begin{align*}
\text{C1: } & 0 \leq s_{i,j}p_{i,j}^{Tx} \leq p_{\text{max}}, \quad \forall i, j, \\
\text{C2: } & \sum_{k=1}^{N_c} \sum_{j=1 \atop k \neq i}^{F_i} s_{k,j}p_{k,j,n}h_{k,j,n}^i \leq I_{th}^{(n)}, \quad \forall i, n, \\
\text{C3: } & s_{i,j} \sum_{n=1}^{N} \log_2(1 + \gamma_{i,j,n}) \geq s_{i,j} R_{\text{min}}, \quad \forall i, j, \\
\text{C4: } & \sum_{j=1}^{F_i} \sum_{n=1}^{N} s_{i,j} \log_2(1 + \gamma_{i,j,n}) \leq R_{\text{Proc}}^{\text{max}}, \quad \forall i,
\end{align*}

By applying the method used in previous section we can formulate the problem as a D.C. programming optimization problem. In other words, similar to (34) and (35) we have:

\begin{align*}
\text{C3: } & Z_{i,j}(\tilde{p}^t) - Q_{i,j}(\tilde{p}^t) \geq R_{\text{min}}, \quad \forall i, j \tag{44} \\
\text{C4: } & Q_i(\tilde{p}^t) - Z_i(\tilde{p}^t) \geq -R_{\text{Proc}}^{\text{max}}, \quad \forall i \tag{45}
\end{align*}
Applying the first order Taylor approximation, the optimization problem can be written as

$$\min_{p} P^{\text{Total}}$$

subject to

$$C1: \ 0 \leq s_{i,j} p^{T_{x}}_{i,j} \leq p_{\text{max}}, \ \forall i, j,$$

$$C2: \ \sum_{k=1}^{N_{c}} \sum_{j=1}^{F_{i}} s_{k,j} p_{k,j,n} h_{k,j,n}^{i} \leq f_{th}^{(n)}, \ \forall i, n,$$

$$C3: \ Z_{i,j}(\tilde{p}^{t}) - \{ Q_{i,j}(\tilde{p}^{t-1}) + \nabla_{p} Q_{i,j}^{T}(\tilde{p}^{t}).(\tilde{p}^{t} - \tilde{p}^{t-1}) \} \geq R_{\text{min}} \ \forall i, j$$

$$C4: \ Q_{i}(\tilde{p}^{t}) - \{ Z_{i}(\tilde{p}^{t-1}) + \nabla_{p} Z_{i}^{T}(\tilde{p}^{t}).(\tilde{p}^{t} - \tilde{p}^{t-1}) \} \geq -R_{\text{Proc}}^{\text{max}} \ \forall i$$

Given sub-channel assignment and power consumption vectors, offloading decisions can be made by users. Recall the power consumption of user $j$ in cell $i$ in (1) and (3). Each user can compare offloading and local processing power consumption to make the decision $s_{i,j}$ as follows:

$$s_{i,j} = \begin{cases} 
1 & \text{if } P_{\text{Local}}^{i,j} > P_{T_{x}}^{i,j} \\
0 & \text{if } P_{\text{Local}}^{i,j} \leq P_{T_{x}}^{i,j}
\end{cases}$$ (47)

IV. ALGORITHM DESIGN

In this section, based on our solutions, we propose two tractable algorithms to solve the optimization problem in a polynomial time. The first algorithm fits well to a situation where information of all cells are available at the centralized unit and base stations are in charge of performing the offloading algorithms. The second algorithm suits well when offloading algorithm is performed at MUs sides and only partial information exchange is required between base stations.
A. J-PAD Algorithm

Algorithm 1 performs Joint Power Allocation and Decision making and is called J-PAD. J-PAD is designed to solve the convex optimization problem presented in (33). Here, the key idea is to make decision and allocate power simultaneously, while channels are assigned beforehand. Algorithm 1 represents the procedure of solving the optimization problem using J-PAD algorithm.

Algorithm 1 Joint Power Allocation and Decision Making (J-PAD) algorithm

1: Initialize power, a, s, \( I_{\text{max}} \), \( \lambda \), and \( \text{Counter} = 0 \)

2: while \( \text{Counter} \leq I_{\text{max}} \) do

3: Channel Allocation

4: Calculate \( EI_{i,j,n} \) based on (16) \( \forall i, j, n \)

5: Form \( a[i] \) based on \( EI_{i,j,n} \)

6: Power Allocation and Offloading Decision

7: for \( i=1 \) to \( N_c \) do

   a) Solve the problem (33) using interior point method [38]

   b) Update Power Vector based on the solution of (33)

   c) Update \( s_{k,u,n} \) according to the solution of (33)

8: end for

9: Update \( \lambda \), \( \text{Counter} = \text{Counter} + 1 \)

10: Centralized unit updates the \( I \) based on (8) and sends this value back to the base stations.

11: end while

J-PAD algorithm is composed of two main sections, channel assignment, based on the equation (16), and power allocation and offloading decision. After performing the second part, the power vectors and offloading decisions are updated at each base station and will be sent to the centralized unit. Then the centralized unit updates the interference value on each channel and sends them back to each base station for next iteration. The problem is solved at the base station where the offloading algorithm is performed.

Besides, \( \lambda \) plays an important role in the performance of J-PAD algorithm. It is a penalty factor to punish the objective function for any value of offloading decision variables, which is not equal to 0 or 1. Therefore \( \lambda \) should be large enough e.g. \( 10^5 \), \( \lambda \gg 1 \) [37], to penalize the
objective. One can fix this value to a predetermined high value but here we first set the $\lambda$ to a relatively low value ($\lambda > 1$). In this case, the value of $s$ will be a real value in $[0,1]$. Then in next iterations we tighten the condition on $s$ by choosing larger $\lambda$.

**B. C-PAD Algorithm**

In this section, we propose an alternative algorithm to J-PAD which has less complexity and the decision making process can be moved to the MU side instead of the BS. In this situation, MUs only need partial information from other cells.

To avoid the integer inherent of the problem, we assign channels and make offloading decision iteratively before allocating the power. Hence, we divide the algorithm into three main parts. 1) Channel allocation which is done based on the (16). 2) Offloading decision which is performed by comparing the alternative solutions power consumption according to the (47). 3) Power allocation. In the latter part, channel allocation and offloading decisions are not optimization variables anymore because they are known for each user beforehand. Therefore, this algorithm performs Channel allocation, Power Allocation and Decision making iteratively and is called C-PAD algorithm. The procedure of finding the solution with C-PAD algorithm is presented in algorithm 2.

In the algorithm 2 the channel allocation scheme is the same as algorithm 1. For offloading section, each user compares its power consumption for two possible cases e.g. local processing or offloading and makes decision accordingly. Given these variables, the problem of power minimization can be solved. This segmentation enables us to perform the algorithm at users’ side. In other words, the second algorithm is a distributed scheme with very low data exchange requirements at the expense of losing optimality. Centralized unit sends information about interference to each base station and the base stations relay this information to the users. Afterwards, users can use their local information and make their decisions. The procedure will continue until the convergence criteria is met. The computational complexity of the proposed algorithms will be discussed and compared in the next section.

**C. Complexity Analysis**

In this section, we investigate the computational complexity of our proposed algorithms. In both J-PAD and C-PAD, to assign sub-channels to the users, we have to find the user with highest effective interference. Let $F$ denote the maximum number of users existing in a cell,
Algorithm 2 Channel allocation, Power Allocation and Decision Making (C-PAD) algorithm

1: Initialize initial points, $I_{\text{max}}$, $\lambda$, and $\text{Counter} = 0$

2: while $\text{Counter} \leq I_{\text{max}}$ do

3: Channel Allocation

4: Calculate $EI_{i,j,n}$ based on (16) $\forall i, j, n$

5: Form $a[t]$ based on $EI_{i,j,n}$

6: Offloading Decision

7: Determine the offloading decision based on (47) for each user.

8: Update the channel allocation and offloading decision vector.

9: Power Allocation

10: for $i=1$ to $N_c$ do

11: a) Solve the problem (28) with a given channel allocation and offloading decision vector using interior point method [38]

12: b) Update Power Vector based on the solution found from (28)

13: end for

14: $\text{Counter} = \text{Counter} + 1$

15: Centralized unit updates the parameter $I$ based on (8) and sends this value back to the base stations and base stations distribute it to the users.

16: end while

i.e., $F = \max_{i=1,...,N_c} F_i$. Since finding the maximum of a set with $K$ elements requires $O(K)$ operations, the sub-channel assignment phase has the complexity order of $O(NF_N)$.

For the data offloading and power allocation in J-PAD algorithm, we have totally $N_cF(N + 1)$ decision variables and $N_c(3F + N + 1)$ convex and linear constraints [37]. Therefore, the computational complexity of solving a joint data offloading and power allocation problem is given by $O((N_cF(N + 1))^3(N_c(3F + N + 1))) \approx O(N_c^4F^3N^3(3F + N))$

In C-PAD algorithm, the data offloading and power allocation are separated. To find the data offloading strategy, it is sufficient to compare the power consumption in cases that each user uses its processor or sends its data to the cloud and select the one with lowest power consumption. Since, we have to carry this out for all users in all cells, we need $O(N_cF)$ operations. For the power allocation, we have totally $N_cFN$ variables and $N_c(2F + N + 1)$ linear and convex
constraints. Similar to what has been presented for the first approach, the power allocation computational complexity has the order of $O(N_c^4 F^3 N^3 (2F + N))$. The computational complexity of proposed methods is summarized in Table II.

**TABLE II: Computational Complexity of proposed approaches.**

|                | Sub-channel Assignment | Data Offloading         | Power Allocation            |
|----------------|------------------------|-------------------------|-----------------------------|
| **J-PAD**      | $O(NF N_c)$            | $O(N_c^4 F^3 N^3 (3F + N))$ |                             |
| **C-PAD**      | $O(NF N_c)$            | $O(N_c F)$              | $O(N_c^4 F^3 N^3 (2F + N))$ |

**V. Simulation Results**

**A. System Parameters**

In this section we evaluate the performance of the proposed algorithms using numerical studies after defining the system parameters and baseline cases. The scenario as depicted in Fig.1 is multi-cell mobile network where each base station is equipped with a computing server. The simulation parameters and their corresponding values are summarized in Table III. We assume that each cell can serve up to $F_i$ users and their QoS is defined as a maximum acceptable delay. The carrier frequency is set to $2GHz$ and thermal noise is considered as a zero mean Gaussian random variable with variance of $\sigma^2$ and power spectral density of $N_0 = -174dbm/Hz$, so $\sigma^2 = (W/N)N_0$. Pathloss model is adopted from [34], shadow fading is modeled as zero mean log normal distribution with variance of $8db$, and Rayleigh fading is modeled as a unit-mean exponential distribution. Each cell has a coverage radius of $500m$ and users are distributed uniformly within a cell coverage.

We have compared our results with two baseline cases to understand the main reasons behind the power savings: whether the saving is more dominated by the offloading decisions or it stems from power control on each channel. In the first one, all MUs use local processing and nobody offloads the data to the cloud. Comparing with this scheme, we can observe how much power saving can be obtained by utilizing the proposed algorithms. Second base line is equal power allocation. In this scheme, power is equally allocated on user’s assigned channels such that required QoS is satisfied. According to the given power allocation, channel assignment is performed based on [16]. Comparing to this scheme, we can find out the amount of power saving related to the power adjustment on each channel.
TABLE III: Simulation parameters values

| Definition                             | Notation | Value          |
|----------------------------------------|----------|----------------|
| Sub-carrier bandwidth                  | $B$      | $200$ $kHz$    |
| Number of sub-carrier                  | $N$      | $25$           |
| Number of cells                        | $N_c$    | $7$            |
| Number of active MUs                   | $F_i$    | $5$            |
| Circuit power                          | $P_c$    | $100$ $mW$     |
| Power amplifier efficiency             | $\eta$   | $0.4$          |
| Scaling factor power                   | $n$      | $3$            |
| Maximum allowable interference         | $I_{th}$ | $-101$ $dbm$   |
| Noise power spectral density           | $N_0$    | $-174$ $dbm/Hz$|
| Maximum transmit power of users        | $P_{max}$| $23$ $dbm$     |
| Maximum delay of user $j$ in cell $i$  | $T_{i,j}$| $100$ $ms$     |
| Average bit stream size of user $j$ in | $L_{i,j}$| $2000$ $bits$  |

![Power Consumption vs. Bit Stream Size](image)

Fig. 2: Aggregate power consumption for different bit stream sizes

B. Simulation Results

The total power consumption of all users over different bit stream sizes is depicted in Fig. 2. The larger bit stream size is, the more power is consumed meanwhile the gap between local computing and proposed algorithms power consumption increases.

Fig. 3 illustrates how J-PAD and C-PAD could help users to offload and how much power
is saved. As can be seen from the figure, by increasing the bit stream size, the percentage of local computing users decreases. The reason is that local processing of the large bit stream size results in higher power consumption in comparison with sending the data to the cloud. Therefore, confirmed by simulations depicted in Fig. 3, users tend to use the alternative option e.g. offloading, to save energy. For large bit stream sizes, using J-PAD and C-PAD, about 30% of users decided local processing and 60% power saving is attained in comparison with local computing base line. Comparing J-PAD and C-PAD, J-PAD slightly outperforms C-PAD in terms of energy saving at most 20%, while it has more complexity.

The maximum acceptable delay as a quality of service requirement is another parameter that affects the power consumption and offloading decisions. In Fig. 4 we have investigated the
delay impact on our algorithms. Longer acceptable delay for offloading users means lower data rate requirement and consequently lower power consumption for sending data. Also for local computing users, it results in lower power consumption confirmed by power model. The gap between the proposed algorithms and the benchmark is wide at the beginning and becomes tighter as maximum acceptable delay gets longer. To discover why, we have illustrated the percentage of the local computing users and the corresponding power savings in Fig. 5. For short delays, the power consumption is relatively high which decays as the acceptable delay becomes longer. It can be seen that users, based on the mobile devices power model, prefer local processing when they can tolerate the considerable delay. Using J-PAD or C-PAD for short delays, about 65% power saving is obtained while for long delays, i.e., when the processing delay requirement can be relaxed, power savings from offloading is diminished because local processing power drops down exponentially with the processing delay due to the power model in Eq. (1), resulting in almost all users processing locally.

Users’ offloading decision not only relies on the power model but also depends on the other users’ decisions due to the interference from the neighboring cells. Consequently, the number of active users in the network is also crucial. Fig. 6 and Fig. 7 address this issue. The more users exist in the network, the higher interference is created which means lower SINR, less data rate and consequently more experienced delay for the users. As a result, the percentage of local users (not necessarily the absolute number) increases with the increasing number of users. For 10 users in cell, proposed algorithms could still achieve about 30% power saving in comparison with local computing base line.

Fig. 5: Power saving vs. percentage of local computing users for different acceptable delays
In Fig 6 we investigate the offloading region for normal and cell edge users to find out when offloading could save power.

For normal users, one can see that for large bit stream size and low acceptable delay, e.g. yellow region in the figures, J-PAD and C-PAD can help mobile devices to save power. For fixed delay, by enlarging the bit stream size we enter to the offloading region. Moreover, C-PAD has a wider region than J-PAD because the offloading decision is made before solving the optimization problem. Our simulation results also reveal that cell edge users with poor channel gain and SINR cannot make benefit from offloading to the cloud. Because with bad channel condition, users need more power than local processing to send data to the cloud with acceptable rate to meet the delay requirements. Here, providing users with better SINR, e.g. using joint transmission, might be helpful.
VI. CONCLUSION

In this paper, the aggregate power consumption of mobile devices as a crucial aspect of mobile cloud computing networks is considered. Accordingly, an optimization problem aimed at minimizing the aggregate power of all users is formulated. To take into account the practical considerations, maximum allowable interference level on each sub-channel and maximum tolerable delay of users are considered. Knowing the inherent non-convexity of our primary problem, we applied the D.C. approximation to transform the non-convex problem to a convex one. We proposed two algorithms, called J-PAD and C-PAD to solve the problem in polynomial time. J-PAD algorithm is better than C-PAD in terms of power saving but with the cost of complexity; therefore, it is not suitable to be used in the mobile terminal but in the BSs with high processing resources. C-PAD has the advantage of running at the users side at the cost of losing the optimality. Our simulations demonstrated that there exist an offloading region for non-cell edge users where they can benefit from offloading data to the cloud. Finally, confirmed by our results, significant enhancement in terms of power consumption of mobile devices could be achieved using proposed algorithms.

REFERENCES

[1] M. Masoudi, B. Khamidehi, and C. Cavdar, “Green cloud computing for multi cell networks,” in 2017 IEEE Wireless Communications and Networking Conference (WCNC) (IEEE WCNC 2017), San Francisco, USA, Mar. 2017.
[2] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[3] J. Kwak, Y. Kim, J. Lee, and S. Chong, “Dream: dynamic resource and task allocation for energy minimization in mobile cloud systems,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.

[4] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, “A framework for cooperative resource management in mobile cloud computing,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.

[5] Y. Zhang, D. Niyato, and P. Wang, “Offloading in mobile cloudlet systems with intermittent connectivity,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.

[6] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[7] M. Claypool and K. Claypool, “Latency and player actions in online games,” *Communications of the ACM*, vol. 49, no. 11, pp. 40–45, 2006.

[8] M. Dusi, S. Napolitano, S. Niccolini, and S. Longo, “A closer look at thin-client connections: statistical application identification for qoe detection,” *IEEE Communications Magazine*, vol. 50, no. 11, 2012.

[9] Y. Xu and S. Mao, “A survey of mobile cloud computing for rich media applications.” *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 1–0, 2013.

[10] P. Piunti, C. Cavdar, S. Morosi, K. E. Teka, E. Del Re, and J. Zander, “Energy efficient adaptive cellular network configuration with qos guarantee,” in *IEEE ICC 2015*, 2015, pp. 1658–1663.

[11] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” *ArXiv e-prints*, Feb. 2017.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “Mobile edge computing: Survey and research outlook,” *CoRR*, vol. abs/1701.01090, 2017. [Online]. Available: [http://arxiv.org/abs/1701.01090](http://arxiv.org/abs/1701.01090)

[13] X. Xu, W. Dou, X. Zhang, and J. Chen, “Enreal: An energy-aware resource allocation method for scientific workflow executions in cloud environment,” *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 166–179, April 2016.

[14] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945–953.

[15] H. Tout, C. Talhi, N. Kara, and A. Mourad, “Smart mobile computation offloading: Centralized selective and multi-objective approach,” *Expert Systems with Applications*, pp. –, 2017.

[16] A. Ellouze, M. Gagnaire, and A. Haddad, “A mobile application offloading algorithm for mobile cloud computing,” in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*. IEEE, 2015, pp. 34–40.

[17] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, “To offload or not to offload? the bandwidth and energy costs of mobile cloud computing,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1285–1293.

[18] X. Liu, Y. Li, and H.-H. Chen, “Wireless resource scheduling based on backoff for multi-user multi-service mobile cloud computing.” 2016.

[19] M.-H. Chen, B. Liang, and M. Dong, “Joint offloading decision and resource allocation for multi-user multi-task mobile cloud,” in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[20] S. E. Mahmoodi, K. Subbalakshmi, and V. Sagar, “Cloud offloading for multi-radio enabled mobile devices,” in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5473–5478.

[21] D. Huang, P. Wang, and D. Niyato, “A dynamic offloading algorithm for mobile computing,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
[22] Y. Im, C. Joe-Wong, S. Ha, S. Sen, M. Chiang et al., “Amuse: Empowering users for cost-aware offloading with throughput-delay tradeoffs,” IEEE Transactions on Mobile Computing, vol. 15, no. 5, pp. 1062–1076, 2016.

[23] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Transactions on Networking, vol. PP, no. 99, pp. 1–1, 2015.

[24] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974–983, 2015.

[25] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, “A framework for partitioning and execution of data stream applications in mobile cloud computing,” ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 4, pp. 23–32, 2013.

[26] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in 2016 IEEE International Symposium on Information Theory (ISIT), July 2016, pp. 1451–1455.

[27] J. Cheng, Y. Shi, B. Bai, and W. Chen, “Computation offloading in cloud-ran based mobile cloud computing system,” in 2016 IEEE International Conference on Communications (ICC). IEEE, 2016, pp. 1–6.

[28] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling,” 2016.

[29] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” IEEE pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[30] Y. Mao, J. Zhang, and K. B. Letaief, “Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems,” CoRR, vol. abs/1701.05055, 2017. [Online]. Available: http://arxiv.org/abs/1701.05055

[31] Y. Liu and M. J. Lee, “An effective dynamic programming offloading algorithm in mobile cloud computing system,” in Wireless Communications and Networking Conference (WCNC), 2014 IEEE. IEEE, 2014, pp. 1868–1873.

[32] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, “Energy aware offloading for competing users on a shared communication channel,” IEEE Transactions on Mobile Computing, vol. 16, no. 1, pp. 87–96, Jan 2017.

[33] A. AL-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, “Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling,” IEEE Transactions on Signal and Information Processing over Networks, vol. PP, no. 99, pp. 1–1, 2017.

[34] E. U. T. R. Access, “Further advancements for e-utra physical layer aspects,” 3GPP TR 36.814, Tech. Rep., 2010.

[35] M. Masoudi, H. Zaeefarani, A. Mohammadi, and C. Cavdar, “Energy efficient resource allocation in two-tier ofdma networks with qos guarantees,” Wireless Networks, pp. 1–15, 2017.

[36] H. H. Kha, H. D. Tuan, and H. H. Nguyen, “Fast global optimal power allocation in wireless networks by local dc programming,” IEEE Transactions on Wireless Communications, vol. 11, no. 2, pp. 510–515, 2012.

[37] E. Che, H. D. Tuan, and H. H. Nguyen, “Joint optimization of cooperative beamforming and relay assignment in multi-user wireless relay networks,” IEEE Transactions on Wireless Communications, vol. 13, no. 10, pp. 5481–5495, 2014.

[38] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.