Rounding Algorithm of Minimizing Executing and Transmitting Time in Multiple CRANs

CHIA-CHENG HU1, WEN-WU LIU1,2, AND JENG-SHYANG PAN3
1College of Artificial Intelligence, Yangtze University, Fuzhou, China
2Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, Zhejiang, China
3College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
Corresponding author: CHIA-CHENG HU (cchu.chiachenghu@gmail.com).

ABSTRACT Recently, Cloud Radio Access Network (CRAN) is a technology for reducing communication load and improving the performance of user tasks. In a CRAN, remote radio heads (RRHs) are densely set up to enable users to connect to the network nearby, and baseband unit (BBU) pools are used to store and execute user contents and tasks. In this paper, our goal is to perform joint optimization when scheduling user tasks and assigning RRH to users in a multiple CRAN infrastructure. The joint problem is NP-hard and modeled as a 0/1 integer linear programming (ILP) with the objective of minimizing the total time of executing user tasks and transmitting user output contents under the constraints of RRH linked capacities and BBU pool computing resources. Then, an algorithm is designed by adopting the rounding technology to solve the 0/1 ILP. We will prove its approximation ratio competitively. The conducted simulations further show that the difference between its solution and the optimal one of the 0/1 ILP is very small, and its approximation ratio is between 1.09 and 1.17. In the conducted experiments, the results demonstrate that the proposed algorithm is indeed practical to be implemented in a multiple CRAN infrastructure.

INDEX TERMS Cloud Radio Access Network (CRAN), resource allocation, task scheduling.

I. INTRODUCTION
Cloud Radio Access Network (CRAN) is a technology in response to explosive growth in data traffic [1][2][3]. It can better support diverse services and deployment requirements through breakthrough innovations such as virtualization, centralization, and programmable [4]. In a CRAN, a hierarchical architecture is adopted by employs two layers of equipment: remote radio heads (RRHs) and baseband unit (BBU) pools [5][6]. To reduce the communication load of the system, users are connected through wireless links by deploying RRHs intensively [7]. Further, the energy of user wireless equipment can be utilized more efficiently due to the short distance from the users to the connected RRHs. RRUs connect to BBU pools through high-speed optical links. The user tasks and contents can be executed and stored in the computing and storing equipment of the BBU pools.

Simultaneously, Software Defined Network (SDN) is another technology and integrated with CRANs to reduce equipment load and system operating costs, and to better control infrastructure [8][9]. In an SDN, the central controller can make the network more intelligent and provide a good platform for the innovation of networks and user applications. By collecting the information of users and user tasks in the entire network, the central controller connects users to the RRHs and schedules user tasks on the servers of BBU pools in a global view. Then, the servers create virtue machines (VMs) to execute the user tasks, and the output contents are transmitted through the wireless links between the users and the connected RRHs.

The system performance is highly related to the capacities of RRHs and BBU pools. The total length of the time period from when the user submits a task to when the output is received is one of the important considerations in determining the system performance of CRAN service. The total length includes the length of executing the submitted task executed and the length of delivering the output to the user. To achieve a shorter total length as possible, the center controller should jointly: 1) offload the submitted task to faster VMs to shorten the execution length, and 2) assign users to nearby RRHs to shorten the delivering length.

Recently, some research works [10-29] focused on similar issues mentioned above. However, most of them solve one of the two problems respectively, and their purpose is to focus on the energy optimization of user equipment. The summary of the above related works is shown in Table I. Different from them, the joint problem of minimizing the
total length of execution and transmitting in the architecture with multiple CRANs (as shown in Fig. 1) is addressed. In Fig. 1, there are \( n \) CRANs. The joint problem of task scheduling and resource allocation are carried out through the coordination and cooperation between a high-level controller and a group of local controllers. In each CRAN, a local controller has a server for creating some VMs to execute the user tasks. Further, it allocates the RRHs in the CRAN for the transmission.

\[
\begin{array}{|c|c|c|}
\hline
\text{Reference} & \text{Infrastructure} & \text{Optimization object} \\
\hline
[10] [11] [12] & CRAN & \text{Min. power consumption} \\
[13] [14] [15] & CRAN & \text{Max. rate among transmitted commodities} \\
[16] & CRAN & \text{Max. network performance} \\
[17] & CRAN & \text{Min. transmission cost} \\
[18] & Cellular network & \text{Min. power consumption} \\
[19] & Cellular network & \text{Min. transmission cost} \\
[20] & Cellular network & \text{Min. transmission time} \\
[21] & FRAN & \text{Min. transmission time} \\
[22] & RAN & \text{Min. bandwidth consumption} \\
[23] & CRAN & \text{Min. transmission cost} \\
[24] & CRAN & \text{Max. spectral efficiency} \\
[25] & CRAN & \text{Min. transmission time} \\
[26] & CRAN & \text{Min. task execution time} \\
[27] [28] & Cloud computing & \text{Min. power consumption} \\
[29] & & \\
\hline
\end{array}
\]

This work Multiple CRAN Min. task execution time and transmission time

Therefore, it is important to schedule tasks by creating VMs and allocating RRHs according to the demand of users among multiple CRANs. To the best of our knowledge, this is the first work to perform joint optimization when scheduling user tasks and assigning RRH to users in the infrastructure with multiple CRANs.

The distinct and major contributions of this paper are listed as follows.

1) We introduce the optimization problem of minimizing the total time of joint task execution and content transmission by scheduling the user tasks on the VMs and allocating the RRHs to the users in the infrastructure with multiple CRANs. The joint problem is NP-hard undoubtedly and formulated as 0/1 integer linear programming (ILP) in this paper.

Due to the NP-hard nature of the joint problem, it is impractical to obtain its optimal solution with exponential computational complexity. Thus, our approach aims to provide a suboptimal solution that can be implemented in a real system. The challenges of proposing an algorithm to obtain this solution include: firstly, it should be proved that the approximate ratio of the proposed algorithm is competitively bounded, and secondly, the execution time of the proposed algorithm must be acceptable in practice.

2) To overcome the above challenges, this paper proposes a new algorithm (named as MTCMC) to Minimize the total time of joint Task execution and Content transmission in Multiple CRANs, which uses the rounding technique is proposed to solve the 0/1 ILP. It produces a linear programming (LP) by relaxing the integer constraints of the
0/1 ILP and utilizes the rounding technique for producing the feasible integer solution of the 0/1 ILP.

II. RELATED WORKS

In [10-16], the works aim to minimize the network power consumption of a CRAN. In [10], this paper considers RRHs allocation and users with multiple antennas under the constraints of user transmitting rate requirements and RRH powers. In [11], the authors consider the problem with the limited fronthaul links in a green CRAN. In [12][13], this paper proposes a transmission mechanism that saves total transmission energy by combining content multicasting and caching.

In [14], the authors propose a holistic framework for green CRAN aiming at minimizing the system power consumption under the constraint of limited fronthaul capacity. In [15], this paper makes full use of the two technologies of mobile cloud computing (MCC) and CRAN to propose an MCC-based CRAN. In [16], by optimizing the beamforming between RRHs and the processor sleeping in BBU pools, the power of CRAN system is minimized with limited forward distance and computational power by utilizing the interaction between communication and computation.

In [17], a cloud-based network is considered, in which heterogeneous base stations (BSs) are connected by routers and backhaul networks with limited capacity wired/wireless links. An algorithm of joint resource allocation and traffic control is proposed to maximize the minimum rate between all transmission commodities. In [18], the authors show that the SDN principle of decoupling control and data planes in RANs can optimize the network performance at coarse timescales, and how fine timescales are made practical in CRANs.

In [19], the authors propose a scheme of caching and multicast in content storing and distributing based on delay requirement in order to minimize the energy consumption of 5G cellular networks. Further, the user requests for the same files are aggregated by multicast stream services. In [20], the authors study the cooperative caching problem of the multi-cell cooperative system from the perspective of minimizing the total cost paid by the content provider. In [21], the authors develop the optimal cooperative content caching and delivery policy to reduce backhaul burden and transmission time.

In [22], the authors aim to maximize the delivery rate is under the constraints of fronthaul capacity and RRH power for fog radio access network (FRAN). In [23], an edge caching algorithm is proposed to minimize the bandwidth consumption of fronthaul and storage costs in FRANs. In [24], the heuristic algorithms are designed with the goal of minimizing the total network traffic cost by jointly allocating system storage and placing monitoring content in the hierarchical cache CRAN architecture of Internet of Things (IoTs) services.

In [25], the authors provide a MIMO technology on optical fiber in tuning an adaptive CRAN system for connecting RRHs. In [26], the services of enhanced mobile broadband and ultra-reliable low-delay communication in 5G are integrated into CRAN to maximize the operator’s revenue. In [27], the work aims to minimize the completion time of parallel jobs composed of a set of independent tasks. They jointly schedule tasks and allocate resources under considering energy consumption.

In [28], an algorithm of optimizing VM’s number in CRANs is proposed by scheduling multiple VMs to a user task and allocating an RRH to multiple users. In [29], the authors attempt to assess the impact of inaccurate information on workflow scheduling due to the available bandwidth of inter-cloud links. Then, a workflow scheduling input mechanism using imprecise information is proposed. In [4], the paper aims to minimize the time length in executing user tasks and transmitting output by optimizing VM scheduling and RRH allocating.

III. PROPOSED ALGORITHM

A. SYSTEM MODEL AND PROBLEM FORMULATION

Consider an infrastructure with \( n \) CRANs, i.e., \( CRAN_{C_1}, CRAN_{C_2}, \ldots, \) and \( CRAN_{C_n} \). Let the set \( C = \{C_1, C_2, \ldots, C_n\} \). In each CRAN, a local controller is responsible for controlling a set of RRHs and a set of BBUs. The RRHs are connected to users for transmitting output contents, and the BBUs create VMs to execute user tasks. We use the sets \( R \) and \( V \) containing RRHs and VMs in \( C \), respectively. Let \( \tau_v \) be the time in second of executing a CPU cycle in \( v \in V \). Each RRH \( r \in R \) is associated to CRAN \( C_r \), and we set \( F^R(v) = C_r \). Each \( r \) can connect up to a limited number (denoted as \( n_r \)) of users. Each \( v \in V \) is generated in \( C_v \), and we set \( F^C(v) = C_v \). Each \( v \) can be scheduled to execute at most \( m_v \) user tasks, and each \( r \) can be connected up to \( m_r \) users.

Further, let the set \( U \) contain the user tasks generated by the users, where each user can only generate a task. Assume that each task \( u \in U \) has the task load and the output content generated after it is executed. The load is the number (denoted as \( n_u \)) of CPU cycles executed to finish the task. The content has the size (denoted as \( s_u \)) in bit, and it can be estimated by the methods of [30]-[33]. The transmission time in second per bit from \( u \) to \( m \), from local controller of \( C_u \) to \( m \) (where \( F^C(r) = C_u \), i.e., \( r \) is associated to \( C_u \)), and from controllers \( C_{c_1} \) to \( C_{c_j} \), are denoted as \( t_{u,m}, t_{c_j,m} \), and \( t_{c_j,c_j} \), respectively.

Let the Boolean variables \( x_{u,v}, y_{u,r} \), and \( z_{a,v,r} \) be the decision variables in the ILP. If \( v \) is generated to execute the task of \( u \) and \( r \) is connected to \( u, x_{u,v} = 1 \) and \( y_{u,r} = 1 \). And, \( z_{u,a,v} = 1 \) if \( F^C(v) \neq F^C(r) \), i.e., \( v \) and \( r \) are belonged to the different CRANs. If the number of tasks is larger than that of the VMs, i.e., \(|U| > |V|\), some tasks will be not scheduled, and no feasible solution exists. Also, if the maximum number of RRH links in a CRAN is smaller than the number of users within the transmission range of the RRHs, some of the users will be not linked.

In order to make that the formulated 0/1 ILP always has a feasible solution, an auxiliary CRAN (denoted as \( C_{n+1} \)) is used. Its RRH links and VMs are enough to link all users and
to be scheduled to execute all user tasks, respectively. Let set \( V_{n+1} (R_{n+1}) \) be the set of the VMs (RRH links) in \( C_{n+1} \). The values of \( t_v, t_{r,v}, t_{c,r} \) and \( t_{c(r)} \) are set to be very large constants which are a lot larger than those in \( C \), respectively. By the setting, the RRH links and the VMs in \( C \) will be used first. The case of user tasks and users scheduled to the VMs in \( V_{n+1} \) and RRH links in \( R_{n+1} \) represents that no VM is allocated to the tasks and no RRH link is connected to the users in this period. The user tasks and their users can be scheduled in next period. All symbols and their meanings are collected in Table II.

| Symbol | Meaning |
|--------|---------|
| Input  |         |
| \( C \) | set of CRANs |
| \( C_c \) | a CRAN and \( C_c \) in \( C \) |
| \( V \) | set of VMs in \( C \) |
| \( R \) | set of RRHs in \( C \) |
| \( t_{n+1} \) | additional CRAN |
| \( V_{n+1} \) | set of VMs in \( R_{n+1} \) |
| \( R_{n+1} \) | set of RRHs in \( C_{n+1} \) |
| \( m_u \) | maximum tasks performed on \( u \) |
| \( m_r \) | maximum users connected to \( r \) |
| \( \tau_p \) | time in second of executing a CPU cycle in \( v \) |
| \( F_C^e(r) \) | CRAN that \( r \) and \( F_C^e(r) \) are associated to |
| \( F_C^e(v) \) | CRAN that \( v \) is generated in and \( F_C^e(v) \) is |
| \( U \) | set of user tasks |
| \( n_u \) | load in CPU cycle of \( u \) |
| \( s_u \) | output content size in bit of \( u \) |
| \( t_{r,v} \) | transmission time in second per bit from \( r \) to \( v \) |
| \( t_{c,r} \) | transmission time in second per bit from local controller of \( C_c \) to \( C_c(r) \) |

Output

| Symbol | Meaning |
|--------|---------|
| \( x_{u,v} \) | \( x_{u,v} = 1 \) if \( v \) is generated to execute the task of \( u \), otherwise \( x_{u,v} = 0 \) |
| \( y_{u,r} \) | \( y_{u,r} = 1 \) if \( r \) is connected to \( u \), otherwise \( y_{u,r} = 0 \) |
| \( z_{u,v,r} \) | \( z_{u,v,r} = 1 \) if \( F_C^e(v) \neq F_C^e(r) \), otherwise \( z_{u,v,r} = 0 \) |

Since each \( u \) in \( U \) is scheduled to one and only one \( v \) in \( V \), the constraint (C.1) of \( \sum_{v \in V} x_{u,v} = 1 \) is derived. The executing time of \( u \) is computed as \( \sum_{v \in V} n_u t_v x_{u,v} \). On the other hand, the time in transmitting the output content is computed as \( \sum_{r \in R} s_u t_{r,v} y_{u,r} \). Due to that one and only one \( r \in R \) will be linked to the associated user of \( u \), the constraint (C.2) of \( \sum_{r \in R} y_{u,r} = 1 \) is derived. Further, the additional transmission time will be induced if \( z_{u,v,r} = 1 \). It is equal to

\[
\sum_{v \in V} \sum_{r \in R} s_u t_{r,v} x_{u,v} + \sum_{r \in R} s_u t_{r,v} y_{u,r} + \sum_{v \in V} \sum_{r \in R} s_u t_{r,v} z_{u,v,r} \tag{1}
\]

Thus, the total time for task execution and content transmission of all tasks is calculated as

\[
\sum_{u \in U} \left( \sum_{v \in V} n_u t_v x_{u,v} + \sum_{r \in R} s_u t_{r,v} y_{u,r} + \sum_{v \in V} \sum_{r \in R} s_u t_{r,v} z_{u,v,r} \right) \tag{2}
\]

The constraint (C.3) is derived from that each \( v \) is scheduled to at most \( m_v \) tasks. In (C.4), the capacity of users connected to the RRHs supported cannot be violated. In (C.5), \( z_{u,v,r} = 1 \) if \( x_{u,v} = 1 \), \( y_{u,v} = 1 \), and \( F_C^e(v) \neq F_C^e(r) \). In (C.7), if the transmission distance of \( r \) cannot reach the user of \( u \), we first set \( y_{u,r} = 0 \). In (C.8), setting \( z_{u,v,r} = 0 \) initially if the user of \( u \) and \( v \) are belonged in the same CRAN, i.e., \( F_C^e(u) = F_C^e(r) \). Finally, the formulated 0/1 ILP is shown as follows.

\[
\sum_{u \in U} \left( \sum_{v \in V} n_u t_v x_{u,v} + \sum_{r \in R} s_u t_{r,v} y_{u,r} + \sum_{v \in V} \sum_{r \in R} s_u t_{r,v} z_{u,v,r} \right) \tag{O}
\]

subject to

\[
\sum_{v \in V} x_{u,v} = 1 \text{ for all } u \in U \tag{C.1}
\]

\[
\sum_{r \in R} y_{u,r} = 1 \text{ for all } u \in U \tag{C.2}
\]

\[
\sum_{u \in U} x_{u,v} \leq m_v \text{ for all } v \in V \cup V_{n+1} \tag{C.3}
\]

\[
\sum_{u \in U} y_{u,r} \leq m_r \text{ for all } r \in R \cup R_{n+1} \tag{C.4}
\]

\[
x_{u,v} + y_{u,r} - z_{u,v,r} \leq 1 \text{ for } F_C^e(v) \neq F_C^e(r), \quad u \in U, \quad v \in V \cup V_{n+1}, \text{ and } r \in R \cup R_{n+1} \tag{C.5}
\]

\[
x_{u,v} \in \{0,1\} \text{ for } u \in U \text{ and } v \in V \cup V_{n+1} \tag{C.6}
\]

\[
y_{u,r} = 0 \text{ if the transmission distance of } r \text{ cannot reach the user of } u, \text{ otherwise } y_{u,r} \in \{0,1\} \text{ for } u \in U \text{ and } r \in R \cup R_{n+1} \tag{C.7}
\]
\[ z_{u,v,r} = 0 \text{ if } F^C(v) = F^C(r), \text{ otherwise } \]
\[ z_{u,v,r} \in [0,1] \text{ for } u \in U, v \in V \cup V_{n+1}, \text{ and } \]
\[ r \in R \cup R_{n+1} \]  \hspace{1cm} (C.8)

B. MTCMC

The algorithm (named as MTCMC) to solve the ILP in the previous section is proposed based on the following three approaches. 1) Since it is difficult to make joint and overall decisions in adjusting two related costs, it is formulated as a 0/1 ILP. 2) Solving the ILP is very time-consuming. Thus, it is transferred into an LP by relaxing its integer decision variables into fractions since the LP is a P problem and its optimal solution is obtained in polynomial time. 3) To satisfy the integer requirements of the 0/1 ILP, it is necessary to round the fractional optimal solution to be the integer one.

The rounding will be performed in the following aspects in MTCMC. If a fractional decision variable is rounded to be an integer, the increment of objective value of the LP will be induced by the rounding. To be as close as possible to the minimum goal of the 0/1 ILP, the fractional decision variable with the smallest increment will be rounded first. The algorithm MTCMC is proposed to determine a feasible solution of the 0/1 ILP by utilizing the rounding technology. Its execution invokes two sub-algorithms, named as INCREMENT and COMPENSATION, where INCREMENT is proposed to compute the objective increment in the rounding process, and COMPENSATION is proposed to compensate the increment. The execution of the two sub-algorithms will be explained later. We describe the execution of MTCMC as follows.

In step 1, we relax the integer decision variables \( x_{u,v,r} \), \( y_{u,v,r} \), and \( z_{u,v,r} \) to fractional variables between zero and one, from the 0/1 ILP. Then, an LP is got, where \( x_{u,v,r} \in [0,1], y_{u,v,r} \in [0,1], \) and \( z_{u,v,r} \in [0,1] \). Its optimal solution of fractional \( x_{u,v,r}', y_{u,v,r}', \) and \( z_{u,v,r}' \), is polynomial time solvable in step 2. In steps 3-28, the fractional \( x_{u,v,r}' \), \( y_{u,v,r}' \), and \( z_{u,v,r}' \), are rounded to \( x_{u,v,r}', y_{u,v,r}' \), and \( z_{u,v,r}' \), which values will be either zero or one, and they will be a feasible solution of the 0/1 ILP. Initially, we set all \( x_{u,v}' = x_{u,v}, y_{u,v,r}' = y_{u,v,r}, \) and \( z_{u,v,r}' = z_{u,v,r} \), in step 3. In steps 4-5, the set (denoted as \( S \)) contains all tuple \((u,v,r)\) for each \( u \in U, v \in V, \) and \( r \in R \). A tuple \((u,v,r)\) \( \in S \) represents a candidate of scheduling \( u \) to \( v \) and linking the user of \( u \) to \( r \).

In step 6, INCREMENT is invoked to compute the increment to the objective value of the problem for each \((u,v,r)\) \( \in S \) if \((u,v,r)\) is adopted for the scheduling and the linking. The increment of \((u,v,r)\) is denoted as \( \Delta_{u,v,r} \).

In step 7, two sets \( \bar{S} \) and \( \bar{S} \) are set to be empty initially, where \( \bar{S} \) will be used for collecting the adopted \((u,v,r)\)s, and \( \bar{S} \) will be used for collecting those not adopted. In steps 8-28, a while-loop is performed for the adoption of \((u,v,r)\)s in \( S \), and its execution is terminated at when \( S \) is empty.

To achieve the aim of the problem, the while-loop in each iteration selects a candidate \((\bar{u}, \bar{v}, \bar{r})\) \( \in \bar{S} \) with the smallest increment \( \Delta_{\bar{u},\bar{v},\bar{r}} \) to the objective in step 9. In step 10, if

\[ |((u,v)|(u,v,r) \in \bar{S})| + 1 \leq m_F, |((u,v)|(u,v,r) \in \bar{S})| + 1 \leq m_V, \text{ and the transmission distance of } \bar{r} \text{ can reach the user of } \bar{u}, (\bar{u}, \bar{v}, \bar{r}) \text{ is adopted by executing steps 11-24. } \]

Otherwise, \((\bar{u}, \bar{v}, \bar{r})\) is not adopted by deleting \((\bar{u}, \bar{v}, \bar{r})\) from \( \bar{S} \) and adding \((\bar{u}, \bar{v}, \bar{r})\) to \( \bar{S} \) in step 26.

In steps 11-28, \((\bar{u}, \bar{v}, \bar{r})\) is adopted by adding \((u,v,r)\) to \( S \), setting \( x_{\bar{u},\bar{v},\bar{r}} = 1 \), and setting \( y_{\bar{u},\bar{v},\bar{r}} = 1 \). In order to satisfy the constraints of (C.1) and (C.2), we set \( x'_{\bar{u},\bar{v}} = 0 \) for \( v \neq \bar{v}, v \in V \cup V_{n+1} \) in step 13, and set \( y_{\bar{u},\bar{r}} = 0 \) for \( r \neq \bar{r}, r \in R \cup R_{n+1} \) in step 14, delete \((u,v,r)\) from \( S \) and \( \bar{S} \) for \( v \in V \cup V_{n+1}, r \in R \cup R_{n+1} \) in step 15, delete \((u,v,r)\) from \( S \) and \( \bar{S} \) for \( u \in U, r \in R \cup R_{n+1} \) in step 16. Further, if \( \bar{v} \) and \( \bar{r} \) belong to the same CRAN, i.e., \( F^C(\bar{v}) = F^C(\bar{r}) \), we set \( x_{\bar{u},\bar{v},\bar{r}} = 0 \) for \( v \in V \cup V_{n+1}, r \in R \cup R_{n+1} \) in step 18. Otherwise, \( x_{\bar{u},\bar{v},\bar{r}} = 1 \) and \( z_{\bar{u},\bar{v},\bar{r}} \) to be \( v \neq \bar{v}, r \neq \bar{r}, v \in V \cup V_{n+1}, r \in R \cup R_{n+1} \), in steps 20-21.

In steps 23-24, the two sub-algorithms COMPENSATION and INCREMENT are invoked. If \( \bar{S} \) is not empty, we schedule \( u \) and link the user of \( u \) for each \((u,v,r)\) \( \in \bar{S} \) to the VMs and the RRHs of the additional CRAN in step 30, respectively.

**MTCMC:**

| Input | Output |
|-------|--------|
| \( C, R, m, V, t, C_{v+1}, V_{n+1}, R_{n+1}, F^C(r), F^C(v) \), \( U, n_u, t_u, t_v, t_{c_l,c_j} \) | \( x'_{u,v}, y'_{u,v}, z'_{u,v,r} \)
1. obtain \( x_{u,v} \in [0,1], y_{u,v} \in [0,1], \) and \( z_{u,v,r} \in [0,1] \) of a LP from relaxing \( x_{u,v} \in [0,1], y_{u,v} \in [0,1], \) and \( z_{u,v,r} \in [0,1] \) of the ILP;
2. obtain \( x_{u,v}, y_{u,v}, \) and \( z_{u,v,r} \) by solving the LP;
3. \( x_{u,v} \leftarrow x'_{u,v}, y_{u,v} \leftarrow y'_{u,v}, \) and \( z_{u,v,r} \leftarrow z'_{u,v,r} \), for \( u \in U, v \in V, \) and \( r \in R \);
4. \( S \leftarrow \{\}; \)
5. add \((u,v,r)\) into \( S \) for each \( u \in U, v \in V, \) and \( r \in R \);
6. execute INCREMENT (\( S \));
7. \( \bar{S} \leftarrow \{\}; \bar{S} \leftarrow \{\}; \)
8. while \( \bar{S} \) is not empty do
9. \( (\bar{u}, \bar{v}, \bar{r}) \leftarrow \arg \min_{(u,v,r) \in S} \Delta_{u,v,r}; \)
10. if \(|((u,v)|(u,v,r) \in \bar{S})| + 1 \leq m_F, |((u,v)|(u,v,r) \in \bar{S})| + 1 \leq m_V, \) and the transmission distance of \( \bar{r} \) can reach the user of \( \bar{u} \)

...
18. $z_{u,v,r} \leftarrow 0$ for $v \in V \cup V_{n+1}, r \in R \cup R_{n+1}$;
19. else
20. $z_{u,v,r} \leftarrow 1$;
21. $z_{u,v,r} \leftarrow 0$ for $v \neq \bar{v}, r \neq \bar{r}, v \in V \cup V_{n+1}, r \in R \cup R_{n+1}$;
22. endif
23. execute COMPENSATION ($\Delta_{u,v,r}$, $S$, $\bar{S}$);
24. execute INCREMENT ($S$, $\bar{S}$);
25. else
26. delete $(u, \bar{v}, \bar{r})$ from $S$; add $(u, \bar{v}, \bar{r})$ into $\bar{S}$;
27. endif
28. endwhile
29. if $S$ is not empty
30. $(u, v, r) \in S$ to the VMs and the RRHs of the additional CRAN;
31. endif

Next, we describe the execution of INCREMENT. When $(u, \bar{v}, \bar{r})$ is adopted, we will set $x_{u,v}^*$ and $y_{u,v}^*$ to be one and the setting induces the increment $\Delta_{u,v,r} = n_t u_t (1 - x_{u,v}^*) + s_{u,v} f_{r,0} (1 - y_{u,v}^*)$ to the objective value. According to the constraints of (C.1) and (C.2), we need to set all $x_{u,v}^* = 0$ for $v \neq \bar{v}, v \in V \cup V_{n+1}$, and set $y_{u,v}^* = 0$ for $r \neq \bar{r}, r \in R \cup R_{n+1}$. Hence, the decrement $\Delta_{u,v,r} = \sum_{v \neq \bar{v}, v \in V \cup V_{n+1}} n_t u_t x_{u,v}^* + \sum_{r \neq \bar{r}, r \in R \cup R_{n+1}} s_{u,v} f_{r,0} y_{u,v}^*$ is induced to the objective value as shown in step 3. Further in step 5, if $FC(u, v, r) = FC(\bar{u}, \bar{v}, \bar{r})$, the decrement $\sum_{v \neq \bar{v}, v \in V \cup V_{n+1}} n_t u_t x_{u,v}^* + \sum_{r \neq \bar{r}, r \in R \cup R_{n+1}} s_{u,v} f_{r,0} y_{u,v}^*$ will be induced by setting $z_{u,v,r} = 0$ for $v \in V \cup V_{n+1}, r \in R \cup R_{n+1}$. In step 7 and step 8, if $FC(u, v, r) \neq FC(\bar{u}, \bar{v}, \bar{r})$, the increment $s_{u,v} f_{r,0} y_{u,v}^*$ will be induced by setting $z_{u,v,r} = 1$ and $z_{u,v,r} \leftarrow 0$ for $v \neq \bar{v}, r \neq \bar{r}, v \in V \cup V_{n+1}, r \in R \cup R_{n+1}$, respectively. In step 10, the net increment $\Delta_{u,v,r} = \Delta_{u,v,r}^+ - \Delta_{u,v,r}^-$.

### COMPENSATION ($\Delta_{u,v,r}, S$, $\bar{S}$):
1. $\omega = \sum_{(u,v,r) \in S \cup \bar{S}} (n_t u_t x_{u,v}^* + s_t u_t y_{u,v}^* + s_{u,v} f_{r,0} c_{(r)} z_{u,v,r})$
2. if $\Delta_{u,v,r} > 0$
3. determine $0 < \alpha < 1$ to meet $(1 - \alpha) \omega = \Delta_{u,v,r}$;
4. $x_{u,v}^* \leftarrow \alpha x_{u,v}^*, y_{u,v}^* \leftarrow \alpha y_{u,v}^*, z_{u,v,r}^* \leftarrow \alpha z_{u,v,r}^*$ for $(u, v, r) \in S \cup \bar{S}$;
5. endif
6. else
7. determine $\alpha > 0$ to meet $\alpha \omega = \Delta_{u,v,r}$;
8. $x_{u,v}^* \leftarrow (1 + \alpha) x_{u,v}^*, y_{u,v}^* \leftarrow (1 + \alpha) y_{u,v}^*, z_{u,v,r}^* \leftarrow (1 + \alpha) z_{u,v,r}^*$ for $(u, v, r) \in S \cup \bar{S}$;
9. endif

### C. ANALYSIS
Based on the following two aspects, we claim that the performance of MTCMC is competitive. First, the obtained
solution of MTCMC is feasible to the 0/1 ILP, that is, the following five conditions should be met: (i) \( \sum_{v \in U} x_{uv}^* = 1 \) for all \( u \in U \), (ii) \( \sum_{r \in R} y_{ur}^* = 1 \) for all \( u \in U \), (iii) \( \sum_{v \in V} x_{uv}^* \leq m \), for all \( v \in V \cup V_{n+1} \), (iv) \( \sum_{v \in V} y_{ur}^* \leq m \), for all \( r \in R \cup R_{n+1} \), and (v) if the transmission distance of \( r \) cannot reach the user of \( u \), \( y_{ur}^* = 0 \). The three conditions (iii), (iv), and (v) are met by the checking in step 10. Lemma 1 will be used for demonstrating that the remaining two conditions are also satisfied. Second, the objective of its solution should have a small difference from the optimal one of the 0/1 ILP. Thus, its approximation ratio will be given in Lemma 2.

Lemma 1. (i) \( \sum_{v \in U} x_{uv} = 1 \) for all \( u \in U \), and (ii) \( \sum_{r \in R} y_{ur} = 1 \) for all \( u \in U \).

Proof. The values of all \( x_{uv}^* \) and \( y_{ur}^* \) are set to either zero or one by executing steps 11-14 or step 30. When \((u, v, r)\) is selected in step 9, steps 12-14 make the parts (i) and (ii) of the lemma to be satisfied. Until the end of executing MTCMC, the two parts will be still satisfied by inserting \((u, v, r)\) into \( S \) in step 11, and delete \((u, v, r)\) from \( S \) and \( T \) in steps 15-16. In step 30, the two parts are always true by scheduling \( u \) and link the user of \( u \) for each \((u, v, r)\) in \( S \) to the VMS and the RRHs of the additional CRAN.

Next, we analyze the approximation ratio of MTCMC. Let \( O_L \), \( O_P \), and \( O_M \) be the optimal objective values of the LP, the problem, and MTCMC, respectively. Undoubtedly, \( O_L \leq O_P \).

Lemma 2. The approximation ratio of MTCMC is defined as \( \frac{O_M}{O_P} \) and it is bounded below by

\[
1 + \frac{\sum_{(u,v,r)} \Delta_u \epsilon_v + \Delta_l \epsilon_r + \Delta_p \epsilon_c(r)}{O_P} \tag{4}
\]

Proof. Let \( \text{iter} \) be the number of iterations executed by the while-loop in MTCMC, and \( O_{M_i} \) as the objective value after the \( i \)th iteration is executed, where \( 1 \leq i \leq \text{iter} \). In each iteration, \((u, v, r)\) is selected in step 9. If it is true after the check in step 10, it is adopted by performing steps 11-24.

And, the increment \( \Delta_{u,v,r} \) is induced to the objective. To compensate \( \Delta_{u,v,r} \), COMPENSATION is invoked in step 23.

In the execution of COMPENSATION, two cases of the compensation are executed in steps 4-5 and 8-9. When \( \Delta_{u,v,r} < \omega \), although the objective is increased by \( \Delta_{u,v,r} \), it is also decreased by \( \Delta_{u,v,r} \) in steps 4-5. When \( \Delta_{u,v,r} > \omega \), the objective is decreased by \( \Delta_{u,v,r} \), it is also increased by \( \Delta_{u,v,r} \) in steps 8-9. Suppose \( i = 1 \). We will have \( O_M_1 = O_L \). Similarly, \( O_M_{i+1} = O_M_i \) when \( 2 \leq i \leq \text{iter} \).

If \( \Delta_{u,v,r} > \omega \) in the \( k \)th iteration and \( 1 \leq k \leq \text{iter} \), steps 4-5 and 8-9 will not be execute and we will have \( O_M_k = O_M_{k-1} + \Delta_{u,v,r} \). Thus, \( O_M = O_M_{k-1} + \sum_{u,v,r} \Delta_{u,v,r} \). When the increment \( \Delta_{u,v,r} \) meets the condition of \( \Delta_{u,v,r} > \omega \), \( \omega \) is almost zero. It is the values of \( x_{uv}^*, y_{ur}^*, \) and \( z_{uvr}^* \) and \( y_{ur}^* \), are also almost zero. Thus, \( \Delta_{u,v,r} = n_u \epsilon_v + s_u \epsilon_r + s_u \epsilon_c(r) \) and \( O_M = 0 \). Due to that \( \Delta_{u,v,r} = \frac{\sum_{(u,v,r)} \Delta_u \epsilon_v + \Delta_l \epsilon_r + \Delta_p \epsilon_c(r)}{O_P} \). Hence, \( O_M/O_P \leq 1 + \sum_{(u,v,r)} \Delta_u \epsilon_v + \Delta_l \epsilon_r + \Delta_p \epsilon_c(r) \).

Starting from the following two points, we believe that in most cases, \( O_M \) and \( O_P \) are very close. That is the gap, i.e., \( \Delta_{u,v,r} \), is very small. 1) In the rounding process of MTCMC, the optimal fractional solution obtained by the LP is taken as the starting point. In most cases, the optimal solutions of the LP and its associative ILP are very close. 2) The fractional decision variables in the LP are rounded to integers in ascending order of their objective increments. The points will be further analyzed and validated in Section IV. In the results of the simulations, the ratio of the gap to \( O_P \) is varied from 0.09 to 0.17.

VI. PERFORMANCE EVALUATION

As far as we know, this is the first work to consider the joint problem in a multiple CRAN architecture. Due to the lack of compared methods, the performance of MTCMC is evaluated through comparison with OPTIMAL and GREEDY, where OPTIMAL is denoted as the optimal solution of the problem and GREEDY is denoted as the solution determined by the algorithm that adopts the greedy strategy in first selecting the task with the minimum length of task execution and content transmission. We obtain OPTIMAL by the branch-and-bound algorithm [34] and use the software tool Lingo [35] for solving OPTIMAL and the LP of MTCMC. OPTIMAL will serve as the benchmarks. Then, we analyze the time complexity of MTCMC and measure the actual execution time of MTCMC.

A. PERFORMANCE COMPARISONS OF MTCMC WITH OPTIMAL AND GREEDY

In the simulations of this section, 16 CRANs are used to implement a multiple CRAN infrastructure. There are 480 static users evenly distributed in the square area covered by the 16 CRANs. Each CRAN covers a circular area, whose radius is 150m, by deploying three RRHs uniformly distributed. It can create up to 50 VMS with evenly distributed CPU cycles between 0.1 and 1. Each RRH can be linked to up to 3 users and each VM can perform up to 3 tasks, where \( n_u \) is evenly distributed between 10³ and 10⁶. The distance between adjacent CRANs is 200m.

We use the method proposed in [36] to compute the transmission time \( \tau_u \) from \( r \) to \( u \) as follows.

\[
\tau_u = \log \left( 1 + \sum_{u \in U, u \in V \cup R} |c_{u,v,r}|^2 / (\epsilon B + \varphi) \right) \tag{5}
\]
where $B = 10$ Mhz is the channel bandwidth, $c_{u,v,p}$ is the channel coefficient and follows independently standard exponential distribution, $p = 1$ W is the transmitting power, $\varepsilon = -169$ dBm/Hz is the noise power spectral density, and $\varphi = 10^{-14}$ is the interference power. We let both of the available bandwidths from the local controllers to the users and between the local controllers are also 10 Mhz. Hence, $t_{c,v} = 10^{-7}$ second and $t_{c,v,p} = 10^{-7}$ second.

In Fig. 2, each VM has the CPU cycles per second between $1 \times 10^{10}$ and $10 \times 10^{10}$. Fig. 2(a) depicts the execution time in VMs while Fig. 2(b) depicts the transmission time in the network. From Fig. 2(a), the execution time decreases with the CPU cycles per second. When the CPU cycles per second decrease, the transmission time increases as shown in Fig. 2(b). The reason is that some of the user tasks are scheduled to the CRANs not belonging to the users. Thus, more traffic load is induced between CRANs. Fig. 2 (c) demonstrates the total time including execution time and transmission, and Fig. 2 (c) demonstrates the approximation ratio of MTCMC. It can be seen from Fig. 2(c) and 2(d) that, compared with the solution of GREEDY, the gap between the solution of MTCMC and the optimal solution of OPTIMAL is smaller. On the other hand, the approximation ratios of MTCMC are between 1.09 and 1.14.

In Fig. 3, we compare the performance of MTCMC with GREEDY and OPTIMAL when the output contents have sizes between 0.5 and 5 Mb. We can observe that with the larger output contents, both execution time and transmission time increase. Further, the approximation ratios of MTCMC are between 1.15 and 1.17.

Finally, in Fig. 4, the number of VMs is between 40 and 58. From Fig. 4(a), we can observe that execution time decreases with VM number increases because more computing resources are supported. And, less transmission time is induced as shown in Fig. 4(b). The approximation ratios of MTCMC are between 1.14 and 1.16.

From Fig. 2 to 4, MTCMC performs much better than GREEDY and very close to OPTIMAL. Its approximation ratios are between 1.09 and 1.17.

![Fig. 2. Time under different CPU cycles per second: (a) execution time, (b) transmission time, (c) total time, and (d) approximation ratio.](image)
Fig. 3. Time under different output content size: (a) execution time, (b) transmission time, (c) total time, and (d) approximation ratio.

Fig. 4. Time under different numbers of VMs: (a) execution time, (b) transmission time, (c) total time, and (d) approximation ratio.

B. TIME COMPLEXITY AND EXECUTION TIME OF MTCMC

There are two main components of executing MTCMC: 1) step 2 for obtaining the optimal solution of the LP; 2) steps
For executing the while-loop. In each iteration of the
while-loop, a content will be determined from the set \( S \)
which contains the contents not scheduled. And, a for-
loop in INCREMENT is executed for computing the
objective increments of those contents in \( S \). Further, the
while-loop is terminated when \( S = \emptyset \). Hence, the time
complexity of the while-loop depends on the number of
contents. Let \( N \) be the number of contents. Thus, the time
complexity of MTCMC is \( O(N^2) \).

As for further evaluating the execution time of solving the
LP, there are one hundred to one thousand contents in the
conducted experiments. Then, two metrics, i.e., executing
time and required memory, are collected. A laptop with an
Intel® Core™i7 2.4 GHz CPU is used to execute the
experiments on the software tool Lingo. In Fig. 5 and 6, the
execution time is between 0.4 and 4.1 seconds and the
required memory is between 3.5 and 35.4 MB. Through the
conducted experiments show that MTCMC is very easy to implement and practical in a real multiple CRAN infrastructure.

V. CONCLUSION
The issue of the joint problem of task scheduling and RRH
link allocation with the objective of minimizing the total time
of executing user tasks and transmitting user output contents
under the constraints of RRH linked capacities and BBU
pool computing resources in a multiple CRAN infrastructure
is advocated. The problem is formulated as a 0/1 ILP, and a
rounding algorithm (named as MTCMC) is designed to solve
the 0/1 ILP. In MTCMC, the optimal fractional solution
obtained by the LP is taken as the starting point. In most
cases, the optimal solutions of the LP and its associative
ILP are very close. Then, the fractional decision variables in
the LP are rounded to integers in ascending order of their
objective increments. The approximation ratio of MTCMC
is bounded in theoretical analysis. In the simulations, the
results show that the gap between the solution of MTCMC
and the optimal solution of the 0/1 ILP is very small. Finally,
the conducted experiments show that MTCMC is very easy
to implement and practical in a real multiple CRAN infrastructure.

REFERENCES
[1] T. Q. S. Quek, M. O. Simeone, and W. Yu, Cloud Radio Access
Networks: Principles, Technologies Applications. Cambridge,
U.K.: Cambridge Univ. Press, 2017.
[2] X. Chen, Z. Han, Z. Chang, G. Xue, H. Zhang, and M. Bennis,
“Adapting downlink power in fronthaul-constrained
hierarchical software-defined RANs,” in Proc. IEEE Wireless
Commun. Netw. Conf. (WCNC), San Francisco, CA, USA, Mar.
2017, pp. 1–6.
[3] L.J. Pu , L. Jiao , X. Chen , L. Wang , Q.Y. Xie, and J.D. Xu,
“Online resource allocation, content placement and request
routing for cost-efficient edge caching in cloud radio access
networks,” IEEE J. Sel. Areas Commun., vol. 36, no. 8, pp. 1751–
1767, Aug. 2018.
[4] W.C. Xia, T.Q.S. Quek, J. Zhang, S. Jin, and H. B. Zhu,
“Programmable Hierarchical C-RAN: From Task Scheduling to
Resource Allocation,” IEEE Trans. Wireless Commun., vol. 18,
no. 3, pp. 2003–2016, Mar. 2019.
[5] R. Tandon and O. Simeone, “Harnessing cloud and edge
synergies: Toward an information theory of fog radio access
networks,” IEEE Commun. Mag., vol. 54, no. 8, pp. 44–50, Aug.
2016.
[6] A. Sengupta, R. Tandon, and O. Simeone, “Fog-aided wireless
networks for content delivery: Fundamental latency trade-offs,”
IEEE Trans. Inf. Theory, vol. 63, no. 10, pp. 6650–6678, Oct. 2017.
[7] K. Wang, W. Zhou, and S. Mao, “On joint BBU/RRH resource
allocation in heterogeneous cloud-RANs,” IEEE Internet Things
J., vol. 4, no. 3, pp. 749–759, Jun. 2017.
[8] M. Zhou, H. Zhang, S. Zhang, L. Song, Y. Li, and Z. Han,
“Design and implementation of device-to-device software-
defined networks,” in Proc. IEEE Int. Conf. Commun. (ICC),
Kuala Lumpur, Malaysia, May 2016, pp. 1–6.
[9] R. Yu, C. Xue, M. Bennis, X. Chen, and Z. Han, “HSDRAN:
Hierarchical software-defined radio access network for
distributed optimization,” IEEE Trans. Veh. Technol., vol. 67,
no. 9, pp. 8623–8636, Sep. 2018.
[10] C. Pan, H. Zhu, N. J. Gomes, and J. Wang, “Joint precoding and
RRH selection for user-centric green MIMO C-RAN,” IEEE
Trans. Wireless Commun., vol. 16, no. 5, pp. 2891–2906, May
2017.
[11] Z. Yu, K. Wang, H. Ji, X. Li, and H. Zhang, “Joint user
association and downlink beamforming for green cloud-RANs
with limited fronthaul,” in Proc. IEEE Global Commun. Conf.
(GLOBECOM), Washington, DC, USA, Dec. 2016, pp. 1–6.
[12] M. Tao, E. Chen, H. Zhou, and W. Yu, “Content-centric sparse
multicast beamforming for cache-enabled cloud RAN,” IEEE
Access.
Trans. Wireless Commun., vol. 15, no. 9, pp. 6118–6131, Sep. 2016.

[13] S. Mosleh, L. Liu, H. Hou, and Y. Yi, “Coordinated data assignment: A novel scheme for big data over cached cloud-RAN,” in Proc. IEEE GLOBECOM, Washington, DC, USA, 2016, pp. 1–6.

[14] K. Guo, M. Sheng, J. Tang, T. Q. S. Quek, and Z. Qiu, “Exploiting hybrid clustering and computation provisioning for green C-RAN,” IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 4063–4076, Dec. 2016.

[15] K. Wang, K. Yang, and C. S. Magarawalage, “Joint energy minimization and resource allocation in C-RAN with mobile cloud,” IEEE Trans. Cloud Comput., vol. 6, no. 3, pp. 760–770, Jul. 2018.

[16] K. Guo, M. Sheng, J.H. Tang, T. Q.S. Quek, and Z.L. Qiu, “On the interplay between communication and computation in green C-RAN with limited fronthaul and computation capacity,” IEEE Trans. Commun., vol. 66, no. 7, pp. 3201–3216, 2018.

[17] W.-C. Liao, M. Hong, H. Farmanbar, X. Li, Z.-Q. Luo, and H. Zhang, “Min flow rate maximization for software defined radio access networks,” IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1282–1294, Jun. 2014.

[18] M. Arslan, K. Sundaresan, and S. Rangarajan, “Software-defined networking in cellular radio access networks: Potential and challenges,” IEEE Commun. Mag., vol. 53, no. 1, pp. 150–156, Jan. 2015.

[19] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, “Exploiting caching and multicast for 5G wireless networks,” IEEE Trans. Wireless Commun., vol. 15, no. 4, pp. 2995–3007, Apr. 2016.

[20] A. Gharabeh, A. Khreishah, B. Ji, and M. Ayyash, “A provably efficient online collaborative caching algorithm for multicell-coordinated systems,” IEEE Trans. Mobile Comput., vol. 15, no. 10, pp. 1863–1876, Aug. 2016.

[21] W. Jiang, G. Feng, and S. Qin, “Optimal cooperative content caching and delivery policy for heterogeneous cellular networks,” IEEE Trans. Mobile Comput., vol. 16, no. 5, pp. 1382–1393, May 2017.

[22] S.-H. Park, O. Simeone, and S. Shitz, “Joint optimization of cloud and edge processing for fog radio access networks,” IEEE Trans. Wireless Commun., vol. 15, no. 11, pp. 7621–7632, Nov. 2016.

[23] X. Wang, S. Leng, and K. Yang, “Social-aware edge caching in fog radio access networks,” IEEE Access, vol. 5, pp. 8492–8501, 2017.

[24] J.J. Yao and N. Ansari, “Joint content placement and storage allocation in C-RANs for IoT sensing service,” IEEE Internet Things J., vol. 6, no. 1, pp. 1060–1067, Feb. 2019.

[25] Y.C. Li, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, “Analogue radio over fiber aided mimo design for the learning assisted adaptive C-RAN downlink,” IEEE Access, vol. 7, pp. 21359–21371, 2019.

[26] J.H. Tang, B.H. Shim, and T. Q. S. Quek, “Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB,” IEEE J. Sel. Areas Commun., vol. 37, no. 4, pp. 881–895, Apr. 2019.

[27] L. Shi, Z. Zhang, and T. Robertazzi, “Energy-aware scheduling of embarrassingly parallel jobs and resource allocation in cloud,” IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 6, pp. 1607–1620, Jun. 2017.

[28] J. Tang, W. P. Tay, T. Q. S. Quek, and B. Liang, “System cost minimization in cloud RAN with limited fronthaul capacity,” IEEE Trans. Wireless Commun., vol. 16, no. 5, pp. 3371–3384, May 2017.

[29] T.A.L. Gneoz, L.F. Bittencourt, N.L.S. da Fonseca, and E.R.M. Madeira, “Estimation of the available bandwidth in inter-cloud links for task scheduling in hybrid clouds,” IEEE Trans. Cloud Comput., vol. 7, no. 1, pp. 62–74, Jan. 2019.

[30] X. Lyu, H. Tian, C. Sengul, and P. Zhang, “Multiuser joint task offloading and resource optimization in proximate clouds,” IEEE Trans. Veh. Technol., vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[31] E. Cuervo et al., “MAUI: Making smartphones last longer with code offload,” in Proc. 8th Int. Conf. Mobile Syst. Appl. Services (ACM MobiSys), San Francisco, CA, USA, Jun. 2010, pp. 49–52.

[32] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proc. USENIX Conf. Hot Topics Cloud Comput., 2010, pp. 4–10.

[33] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, “A framework for partitioning and execution of data stream applications in mobile cloud computing,” in Proc. IEEE 5th Int. Conf. Cloud Comput., Honolulu, HI, USA, Jun. 2012, pp. 794–802.

[34] A. Geoffrion and R. Marsten, “Integer Programming Algorithms: A Framework and State-of-the-Art Survey,” Management Science, vol. 18, pp. 465–491, 1972.

[35] Lingo, http://www.lindo.com.

[36] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, “Cache placement in FogRANs: From centralized to distributed algorithms,” IEEE Trans. Wireless Commun., vol. 16, no. 11, pp. 7039–7051, Nov. 2017.