Joint Computation and Communication Resource Allocation for Energy-Efficient Mobile Edge Networks

Johnson Opadere*, Qiang Liu*, Ning Zhang† and Tao Han*
*Electrical and Computer Engineering Department,
University of North Carolina at Charlotte, NC, United States
†Computer Science Department, Texas A&M University at Corpus Christi, TX, United States
Email: *{jopadere, qliu12, Tao.Han}@uncc.edu; †ning.zhang@tamucc.edu

***Abstract*—In this paper, an ultra-dense mobile edge network is studied, where base stations (BSs) are equipped with computation resources to execute users' offloaded tasks. Although an ultra-dense BS deployment provides seamless coverage and reduced computation latency of the offloaded tasks, the cost of network power consumption is increased. We formulate an optimization problem to jointly optimize active BSs set, uplink and downlink beamforming vector selection, and computation resource allocation in order to tackle the power consumption and latency trade-off. To efficiently solve this problem, we propose a sequential solution framework. Specifically, we first select the active BSs based on communication and computation power-aware selection rule. The computation resources and dual-link beamformers are subsequently optimized for the satisfaction of task computation deadline, network energy savings and improved coverage. Simulation results show that the proposed joint optimization framework significantly reduces the network power consumption.**

***Index Terms*—Mobile edge computing; Base station sleep-mode; Computation resource optimization**

I. Introduction

Mobile traffic is anticipated to grow dramatically, due to the continued proliferation of mobile devices. The fifth generation (5G) wireless network is required to significantly increase network capacity to accommodate the massively growing mobile traffic [1]. One of the fundamental techniques for boosting network capacity is network densification, where base stations are deployed in an ultra-dense fashion. Also, incorporation of cloud computing into radio access networks (RANs), facilitated by Cloud-RAN, is geared towards meeting the requirement of system capacity efficiently. Cloud-RAN is based on centralization of base station baseband processing into a pool with data traffic conveyed between the pool and densely deployed remote radio heads (RRHs) over a fronthaul.

However, Cloud-RAN and network densification come with some challenges, such as fronthaul constraints [2] and increase in energy budget due to the constant operation of the network devices [3]. To alleviate these constraints, mobile edge and fog-computing based RANs are proposed to alleviate the Cloud-RAN fronthaul challenge. These computing based RAN architectures move cloud services such as computing and storage to the RRHs, closer to the user equipment (UEs). Depending on the architecture and the functionalities, the computation equipped RRHs have been termed enhanced RRHs (eRRHs) [4], fog-computing Access Points (F-APs) [5], and edge-computing Access Points (EC-APs) [6]. The term Access Points (APs) is adopted in this paper to describe these base stations capable of supporting radio and edge node computing services. That is, the functionalities at the AP include the capabilities of the radio frequency, the physical layer, and the upper layers procedures.

With edge and fog computing, the densely deployed APs can provide radio resource to the UEs, and capability of execution of computation tasks. However, such AP network deployment exhibits significant traffic variations [3]. In a trace study conducted in [3], the traffic profile significantly differs between weekdays and weekends. Hourly fluctuation of traffic is also observed in the same study. Consequently, some APs are either idle or underutilized at low traffic. Higher fluctuating traffic profile is expected in 5G wireless systems, where densely deployed edge devices, such as APs, can be idle at every time instant [7]. Therefore, it will be considerably energy-efficient to put the optimal number of APs into operation to meet the communication and computation demands of the UEs and switch off other APs.

A. Contributions

The main contributions of this paper are summarized in the following.

- We jointly optimize both uplink and downlink beamforming vectors, the edge server CPU cycles and active AP subset. To the best of our knowledge, this is the first work that jointly considers the dual link beamforming vectors, CPU cycles, and optimal AP selection.
- We select optimal APs based on holistic energy selection rule. The power consumption rule encompasses computation processing power compared to the ubiquitous radio resources only power consumption.
- We evaluate the performance of the proposed algorithm. Simulation results show considerable power savings of our proposed scheme as compared to the relevant benchmarks.

B. Related Work

Although energy efficient wireless communication networks have been extensively studied [8]–[10], there exist only a few works on energy efficient computation at the edge servers [11]–[14]. The works like [13], [14] focus on optimizing resource allocation for a reduction in AP power consumption. However, optimal selection of AP is not considered in these works. While [15] considers optimal AP mode that will yield efficient power consumption, beamforming vectors are not included in the optimization strategy. AP downlink energy transmit beamformer is jointly optimized with CPU frequencies in [16], but the sleep-mode strategy is not considered, thus making it inefficient for the ultra-dense network if all APs are active at all times regardless of traffic fluctuations.

In contrast to the above works, this paper jointly considers reducing network system power dissipated by communication and computation infrastructure and resources by selecting an optimal number of active APs and minimizing the dual link beamforming vectors, and optimally allocating computation resource for offloaded tasks processing at the network edge.

II. System Model

Without loss of generality, we consider a densely deployed edge computing cellular network based on C-RAN architecture. The network consists of I identical RRHs equipped with N antennas each. Each RRH is endowed with a computation server. Thus, the coupling of RRH-server is referred to as AP as noted earlier. The APs set in the region is indexed by $\mathcal{I} = \{1, 2, ..., I\}$. Let $\mathcal{L} \subseteq \mathcal{I}$ denote the set of active APs and $\mathcal{Z}$ denote the set of APs in sleep mode, such that $\mathcal{L} \cup \mathcal{Z} = \mathcal{I}$. Each UE is equipped with a single antenna. The UEs set in the region is indexed by $\mathcal{K} = \{1, 2, ..., K\}$. Due to limited computational capacity, each UE processes a fraction of its computation task locally and offload the remaining to its associated AP for processing.

A. Communication Model

We consider the uplink and downlink transmission in the model. Let the vector consisting of the channels from all the APs to the k-th UE be $\mathbf{h}_k^{\mathsf{H}} = \left[\mathbf{h}_{1k}^{\mathsf{H}}, \cdots, \mathbf{h}_{Ik}^{\mathsf{H}}\right]$. Channel reciprocity is assumed in the uplink and downlink channels such that with time-division duplex (TDD), the channel vector in the uplink is merely the transpose of that in the downlink [17].

For the k-th UE uploading its computation task to the AP, we denote the uplink transmission normalized symbol as $s_k^U \in \mathbb{C}$, such that $\mathbb{E}\left[|s_k|^2\right] = 1$. All UEs are assumed to transmit with an identical power p^U. Thus, the transmit signal from the UE k is

$$x_k^U = \sqrt{p^U} s_k^U, \quad \forall k \in \mathcal{K}. \tag{1}$$

Therefore, the received uplink signal at all APs from the k-th UE is

$$\mathbf{y_k} = \sum_{k\in\mathcal{K}} \mathbf{h}_{ik}\sqrt{p^U} s_k^U + \boldsymbol{\eta}_k^U, \tag{2}$$

where $\boldsymbol{\eta}_k^U \in \mathbb{C}^N$ is the receiver noise vector at all APs consisting of circularly symmetric complex Gaussian random variables each distributed as $\mathcal{CN}\left(0, \sigma^2\right)$.

Let $\mathbf{m}_k^U$ denote the receiver beamforming vector used to decode s_k^U from the kth UE. The signal-to-interference-plus-noise ratio (SINR) of the k-th UE uplink transmission after applying $\mathbf{m}_k^U$ is given by

$$\gamma_k^U = \frac{p^U \left|\left(\mathbf{m}_k^U\right)^T \mathbf{h}_k\right|^2}{\sum_{k\in\mathcal{K}, j\neq k} p^U \left|\left(\mathbf{m}_k^U\right)^T \mathbf{h}_j\right|^2 + \left\|\mathbf{m}_k^U\right\|_2^2 \sigma^2}, \tag{3}$$

where $\mathbf{m}_k^U = \left[\left(\mathbf{m}_{1k}^U\right)^T, ..., \left(\mathbf{m}_{Ik}^U\right)^T\right]^T$. Also, $\mathbf{m}_{ik}^U \in \mathbb{C}^N$ represents the beamforming vector for the k-th UE received at the i-th AP, where the array of RRH antennas acts as the receiver for the K independent streams of data transmitted from the UEs. The uplink achievable rate for k-th UE is expressed as

$$R_k^U = W \log_2(1 + \gamma_k^U). \tag{4}$$

Let $\mathbf{w}_{ik}^D \in \mathbb{C}^N$ denotes the downlink transmission beamforming vector from AP i to UE k and the downlink transmitted normalized symbol denoted by s^D. The transmitted signal is given as

$$x_i^D = \sum_{k\in\mathcal{K}} \mathbf{w}_{ik}^D s_k^D, \quad \forall i \in \mathcal{I}. \tag{5}$$

The aggregated beamforming vectors from all APs to the k-th UE is therefore denoted as $\mathbf{w}_k^D = \left[\left(\mathbf{w}_{1k}^D\right)^T, ..., \left(\mathbf{w}_{Ik}^D\right)^T\right]^T$. Thus, we can write the received signal $y_k^D \in \mathbb{C}^N$ by the k-th UE as

$$y_k^D = \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k^D s_k^D + \sum_{j\neq k}^{K} \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j^D s_j^D + \eta_k^D, \tag{6}$$

where $\eta_k^D \sim \mathbb{CN}\left(0, \sigma^2\right)$ is the additive white Gaussian noise at the k-th UE . The downlink signal-to-interference-plus-noise ratio (SINR) for the UE k can be expressed as

$$\Upsilon_k^D = \frac{\left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k^D\right|^2}{\sum_{k\in\mathcal{K}, j\neq k} \left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j^D\right|^2 + \sigma^2}. \tag{7}$$

The downlink achievable rate of the k-th UE is therefore

$$R_k^D = W \log_2(1 + \Upsilon_k^D). \tag{8}$$

B. Computation Model

We consider a computational model where the UE executes a fraction of its computation locally and offload the remaining [18]. Each UE offloaded portion of its task to its associated AP is described by tuple $\langle D_k, U_k, T_k \rangle$. D_k denotes the size of the input data (in bits) of the computation task from the k-the UE to the i-th AP, which may include program codes and input parameters. U_k represents the total number of AP server CPU cycles required to complete the task, and T_k denotes the task completion deadline (in seconds). Each task is atomic and cannot be partitioned into subtasks; hence a task cannot be offloaded to more than one AP.

We can compute the transmission time of k-th UE for offloading the task size D_k to the i-th AP as

$$T_{i,k}^U = \frac{D_k}{R_k^U}. \tag{9}$$

1) AP Server Computing Cost

The total energy consumption at the i-th AP due to CPU computation execution, denoted by E_i^{comp} is given as [7]

$$E_i^{comp} = \sum_{k\in\mathcal{K}} \kappa U_k f_{i,k}^2, \tag{10}$$

where κ is a hardware architecture related constant. $f_{i,k}$ is the computation capacity in cycles per second of the AP server allocated to k-th UE's task. The computation at each server is limited by $\sum_{k\in\mathcal{K}} f_{i,k} \leq F_i^{max}$, which implies that the maximum computation capacity in cycles per second of i -th AP server is denoted by F_i^{max}. The overhead in terms of time for executing the k-th UE's task by the i-th AP server is expressed as

$$T_{i,k}^{exe} = \frac{U_k}{f_{i,k}}. \tag{11}$$

2) Computation latency

Similar to [19] we ignore the latency for computation result delivery from AP server to the UEs because, in many applications, the computation outcome is much smaller than the input task size [20]. The total remote computation latency for the offloaded k-th UE's task at the i-th AP is therefore

$$T_{i,k}^{tot} = T_{i,k}^{U} + T_{i,k}^{exe}. \tag{12}$$

C. Power Consumption Model

1) Communication Power Consumption Model

Here, we consider the power consumption of a conventional BS (excluding the computation server). To make a distinction between the power consumed by the conventional BS components and the added server, we term the power consumed by the former as *communication power*, and the power consumption of the latter as *computation power*.

We represent the operation power (excluding transmission power) of the i-th AP and its wired fronthaul link while on the active mode as P_i^{active}. The power consumed by the i-th AP and its fronthaul link on the sleep-mode is denoted by P_i^{sleep}. Therefore, the total communication power consumption can be expressed as

$$P_{comm} = \eta^{-1} \sum_{i \in \mathcal{L}} P_{tr} + \sum_{i \in \mathcal{L}} P_i^{active} + \sum_{i \in \mathcal{Z}} P_i^{sleep}.. \tag{13}$$

Since $\mathcal{L} \cup \mathcal{Z} = \mathcal{I}$, then $\sum_{i\in\mathcal{Z}} P_i^{sleep} = \sum_{i\in\mathcal{I}} P_i^{sleep} - \sum_{i\in\mathcal{L}} P_i^{sleep}$. Also, substituting $\sum_{i\in\mathcal{L}} P_i^{d} = \sum_{i\in\mathcal{L}} P_i^{active} - \sum_{i\in\mathcal{L}} P_i^{sleep}$, and $P_i^{tr} = \|\mathbf{w}_{ik}\|_2^2$, the communication power can be rewritten as

$$P_{comm} = \eta^{-1} \sum_{i \in \mathcal{L}} \|\mathbf{w}_{ik}\|_2^2 + \sum_{i \in \mathcal{L}} P_i^{d} + \sum_{i \in \mathcal{I}} P_i^{sleep}, \tag{14}$$

where η is the RF power efficiency, which depends on the number of transmitter antenna [21].

2) Computation Power Consumption Model

Using (10) and (11), we can compute the total computation power at AP server i as

$$P_i^{comp} = \sum_{k \in \mathcal{K}} \frac{\kappa U_k f_{i,k}^2}{T_{i,k}^{exe}} = \sum_{k \in \mathcal{K}} \kappa f_{i,k}^3. \tag{15}$$

Thus, the network computation power is

$$P_{comp}\left(\mathbf{f}\right) = \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{K}} \kappa f_{i,k}^3. \tag{16}$$

It can be seen that in AP sleep-mode P_i^{comp} is 0, as only AP $i \in \mathcal{L}$ is involved in workload computation. Thus we propose putting the server into the sleep-mode with the RRH for optimal power savings.

3) Total Network Power

The total network power consumption can now be aggregated as $P_{tot} = P_{comm}\left(\mathcal{L}, \mathbf{w}\right) + P_{comp}\left(\mathbf{f}\right)$, explicitly as

$$P_{tot} = \sum_{i \in \mathcal{L}} \left[\eta^{-1} \sum_{k \in \mathcal{K}} \|\mathbf{w}_{ik}\|_2^2 + P_i^d + \sum_{k \in \mathcal{K}} \kappa f_{ik}^3 \right]. \tag{17}$$

Since the last term $\sum_{i\in\mathcal{I}} P_i^{sleep}$ of (14) is a constant involving all APs, it is trivial to optimal active AP selection problem, and it is therefore omitted in (17).

III. Problem Formulation

In times of low traffic, an optimal number of active APs can be sought. The remaining APs are put into the sleep-mode, while their associated UEs are transferred to the active APs for communication and remote computation support. This is illustrated in Fig. 1.

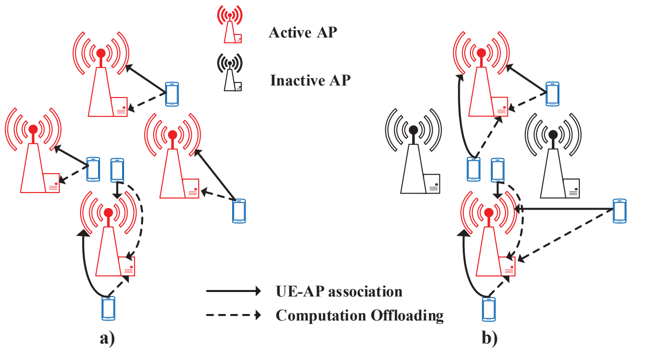

Fig. 1. Communication and computation UEs support with active AP set selection. a) All APs are active. b) Some APs are put into the sleep-mode, and their UEs transferred to the active APs for communication and computation task processing.

The optimization problem can be formulated as

$$\begin{aligned} \mathscr{P}: \quad & \min_{L,w,m,f} \quad P_{tot} \\ & \text{s.t.} \quad \mathcal{C}1: \sum_{k} \|\mathbf{w}_{ik}\|_2^2 \leq P_i^{max}, \quad \forall i \in \mathcal{L} \\ & \qquad \mathcal{C}2: \sum_{k} \|\mathbf{w}_{ik}\|_2^2 = 0, \quad \forall i \in \mathcal{Z} \\ & \qquad \mathcal{C}3: R_k^U \geq R_{k,min}^U, \quad \forall k \in \mathcal{K} \\ & \qquad \mathcal{C}4: R_k^D \geq R_{k,min}^D, \quad \forall k \in \mathcal{K} \\ & \qquad \mathcal{C}5: \sum_{k \in \mathcal{K}} f_{ik} \leq F_i^{max}, \quad \forall i \in \mathcal{L} \\ & \qquad \mathcal{C}6: \sum_{k \in \mathcal{K}} f_{ik} = 0, \quad \forall i \in \mathcal{Z} \\ & \qquad \mathcal{C}7: T_{i,k}^{tot} \leq T_{i,k}^{max} \end{aligned} . \tag{18}$$

where $\mathbf{f} = \left[\mathbf{f}_1^T, ..., \mathbf{f}_K^T\right]^T$, $\mathbf{f}_k = \left[f_{1,k}, ..., f_{I,k}\right]$ and $\mathbf{w} = \left[\mathbf{w}_{1k}^{\mathsf{H}}, \mathbf{w}_{2k}^{\mathsf{H}}, \cdots, \mathbf{w}_{Ik}^{\mathsf{H}}\right]^{\mathsf{H}}$. $\mathcal{C}1$ gives the upper limit of the AP transmit power. $\mathcal{C}2$ enforces $\mathbf{w}_{ik} = 0$ for any k-th UE not associated with any i-th AP . The uplink and downlink minimum data rates are given in $\mathcal{C}3$ and $\mathcal{C}4$, respectively. $\mathcal{C}5$ gives the upper limit of CPU cycles to be allocated to the computation tasks by the AP, and $\mathcal{C}6$ ensures the computation power of an AP having no offloaded tasks is 0. The total allowable deadline for each offloaded task is given in $\mathcal{C}7$.

The uplink and downlink data rates constraints of problem $\mathscr{P}$ are non-convex. First, we rewrite the downlink data rate constraint into second-order cone equivalent [22].

$$\mathcal{C}3\text{a:}\quad \|\mathrm{r}_k^D\|_2 \leq \sqrt{1+1/(2^{R^D_{k,\min}/W}-1})\text{Re}\{u_{kk}\}, \forall k, \tag{19}$$

where $r_k^D = [u_{k1}, u_{k2}, ..., u_{kk}, \sigma_k]^T$ $u_{kj} = \sum_{i \in \mathcal{L}} \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j$.

For the uplink transmission rate constraint, we transform the uplink channel into a virtual downlink channel using uplink and downlink duality. The duality implies that same minimum transmit power is required to meet the downlink SINR target as in the uplink target, where the uplink channel is actualized by downlink's input and output reversal [23]. After application of uplink and downlink duality, uplink beamforming vectors $\mathbf{m}_k^U$ can be rewritten as $\mathbf{v} = \left[(\mathbf{v}_{1k})^T, ..., (\mathbf{v}_{Ik})^T\right]^T$, where $\mathbf{v}_{ik} \in \mathbb{C}^{IN}$. Thus, the second-order cone form is

$$\mathcal{C}4\text{a:}\quad \|\mathrm{r}_k^U\|_2 \leq \sqrt{1+1/(2^{R^U_{k,\min}/W}-1})\text{Re}\{d_{kk}\}, \forall k, \tag{20}$$

where $r_k^U = [d_{k1}, d_{k2}, ..., d_{kk}, \sigma_k]^T$,
$d_{kj} = \sum_{i \in \mathcal{L}} \mathbf{h}_k^{\mathsf{H}} \mathbf{v}_j$. The optimization problem becomes

$$\begin{aligned} \mathscr{P}1.1:\quad & \min_{L,w,v,f} \quad P_{tot} \\ & \text{s.t.} \quad \mathcal{C}1, \mathcal{C}2, \mathcal{C}3\text{a}, \mathcal{C}4\text{a}, \mathcal{C}5-\mathcal{C}7. \end{aligned} \tag{21}$$

IV. Problem Analysis

Despite the transformation to yield second-order cone (SOC) form in $\mathcal{C}3$ and $\mathcal{C}4$, the problem $\mathscr{P}1.1$ is still not convex due to the latency constraint, $\mathcal{C}7$. Using (9), (11) and (12), $\mathcal{C}7$ can be rewritten as

$$\frac{D_k}{R_k^U} + \frac{U_k}{f_{i,k}} \leq T_{i,k}^{max}. \tag{22}$$

Therefore $\mathcal{C}7$ is reformulated as

$$\mathcal{C}7\text{a:}\quad R_k^U \geq \frac{D_k}{T_{i,k}^{max} - \frac{U_k}{f_{i,k}}}. \tag{23}$$

Yet, (23) is not tractable because it is embedded with two decision (optimization) variables $\mathbf{v}$ in R_k^U, and $f_{i,k}$. Thus, we start out by fixing $f_{i,k}$ and optimizing other variables. One way to fix $f_{i,k}$ is to allocate it based on its U_k relative to all others at the AP. Let each U_k of D_k data arrived at AP i be denoted by U_{ik}, the number CPU cycles allocated by AP i for task k is

$$\widetilde{f}_{i,k} = \frac{U_{ik}}{\sum_{k \in \mathcal{K}} U_{ik}} F_i^{max}, \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{L}. \tag{24}$$

Similar to (20), $\mathcal{C}7a$ can be transformed to so SOC form

$$\mathcal{C}7\text{b:}\quad \|\mathrm{r}_k^U\|_2 \leq \sqrt{1+1/(2^{\delta_k/W}-1})\text{Re}\{d_{kk}\}, \forall k, \tag{25}$$

where $\delta_k = \frac{D_k}{T_{ik}^{max} - \frac{U_k}{\widetilde{f}_{i,k}}}$. With fixed $\widetilde{f}_{i,k}$, the problem is restated as

$$\begin{aligned} \mathscr{P}1.2:\quad & \min_{L,w,v} \quad P_{tot} \\ & \text{s.t.} \quad \mathcal{C}1, \mathcal{C}2, \mathcal{C}3\text{a}, \mathcal{C}4\text{a}, \mathcal{C}5, \mathcal{C}6, \mathcal{C}7b. \end{aligned} \tag{26}$$

To solve problem $\mathscr{P}1.2$, we consider a case of a given active APs set $\mathcal{L}$. Hence, the AP transmission power constraint $\mathcal{C}1$ can be rewritten as

$$\mathcal{C}1a:\quad \sum_k \|\mathbf{L}_{ik}\mathbf{w}_{ik}\|_2^2 \leq P_i^{max}, \quad \forall i \in \mathcal{L}, \tag{27}$$

where $\mathbf{L}_{ik} \in \mathbb{C}^{LN \times LN}$ is a block diagonal matrix having i-th main diagonal identity matrix $\mathbf{I}_N$ and zeros elsewhere. With a given active AP set $\mathcal{L}$, the problem can be restated as

$$\begin{aligned} \mathscr{P}1.3\left(\mathcal{L}\right):& \min_{w,v} \quad \sum_{i \in \mathcal{L}} \left[\eta^{-1} \sum_{k \in \mathcal{K}} \|\mathbf{L}_{ik}\mathbf{w}_{ik}\|_2^2 + P_i^d + \sum_{k \in \mathcal{K}} \kappa \widetilde{f}_{i,k}^3\right] \\ & \text{s.t.} \quad \mathcal{C}1\text{a}, \mathcal{C}2, \mathcal{C}3\text{a}, \mathcal{C}4\text{a}, \mathcal{C}5, \mathcal{C}6, \mathcal{C}7b. \end{aligned} \tag{28}$$

It is possible to solve problem $\mathscr{P}1.3\left(\mathcal{L}\right)$ by the interior method, and subsequently search over all AP sets that will minimize network power consumption, but this approach will be computationally expensive [24]. Instead, we will apply AP selection algorithm similar to the greedy algorithm to select optimal active AP set.

A. Active AP selection

The selection rule is akin to the *minimum-increase rule* in the successive thinning algorithm in [25] for sparse filter design applied for backward selection in [24]. In contrast to our holistic power computation rule, the selection in [24] is limited to the communication resource. Hence, it is inapplicable to delay-sensitive networks. With the set of inactive APs initially set to null, the problem $\mathscr{P}1.3$ is iteratively solved and the AP that yields maximum total power consumption reduction, when switched off, in each iteration is removed from the active set and added to the inactive set until the optimal active set is achieved. At each iteration, while the problem is feasible, $\mathbf{w}$, $\mathbf{v}$, and $\mathbf{f}$ are re-optimized for the remaining set of APs. We assume the feasible region of $\mathscr{P}1.3\left(\mathcal{L}^{[j]}\right)$ is nonempty. As shown in Algorithm 1, $y^{[j]}$ denotes the AP at iteration j, obtained as $y^{[j]} = \underset{i \in L^{[j]}}{\arg\min} \mathscr{P}1.3\left(\mathcal{L}^{[j]}\right)$ that yields minimum power consumption when its AP is switched off and thereafter added to the inactive set $\mathcal{Z}^{[j+1]}$. The removal in the procedure is without replacement. The optimal active AP set is denoted by $\mathcal{L}^*$, and we depict the optimal $\mathbf{w}$, and $\mathbf{v}$ of this procedure as $\mathbf{w}^*$, and $\mathbf{v}^*$, respectively.

Algorithm 1: Active AP Set Selection Algorithm

```
1 Initialize $j = 0$, $\mathcal{L}^{[0]} = \{1, ..., I\}$ and $\mathcal{Z}^{[0]} = \emptyset$.
2 With $\mathbf{f} = \widetilde{\mathbf{f}}$, solve problem $\mathscr{P}1.3\left(\mathcal{L}^{[j]}\right)$, and obtain
    $\mathscr{P}^*1.3\left(\mathcal{L}^{[j]}\right)$
3 if feasible then
4     Solve $y^{[j]} = \underset{i \in L^{[j]}}{\arg\min} \mathscr{P}1.3\left(\mathcal{L}^{[j]}\right)$. Update
      $\mathcal{Z}^{[j+1]} = \mathcal{Z}^{[j]} \cup y^{[j]}$, and $\mathcal{L}^{[j+1]} = \mathcal{L}^{[j]} / y^{[j]}$,
      $j \leftarrow j + 1$, and go to step 3.
5 end
6 else
7     Go to Step 9
8 end
9 Output optimal AP active set $\mathcal{L}^{*[j]}$ if $j = 0$ or $\mathcal{L}^*[j-1]$ if
    $j \geq 1$, and the optimal $\mathbf{w}^*$, and $\mathbf{v}^*$.
```

B. Joint Optimization Algorithm

To achieve efficient network power consumption, joint optimization of the AP active set, the uplink and downlink beamforming vectors, and the AP allocated CPU cycles per task is thereafter implemented. With optimal number of APs realized from Algorithm 1, $\mathcal{L}^*$, $\mathbf{w}^*$, and $\mathbf{v}^*$ are obtained using $\mathscr{P}1.3\,(\mathcal{L})$. The CPU cycles are then unfixed, and the optimal number of cycles is obtained by solving $\mathscr{P}1.1\,(\mathcal{L}^*, \mathbf{w}^*, \mathbf{v}^*)$. The overall approach is given in Algorithm 2.

Algorithm 2: Overall Algorithm for the Joint Optimization problem

```
Transform constraints $\mathcal{C}3$, $\mathcal{C}4$ and $\mathcal{C}7$ using (19), (20) and
  (25), respectively
Fix $\mathbf{f} = \tilde{\mathbf{f}}$ using (24).
Solve $\mathscr{P}1.3\,(\mathcal{L})$ using Algorithm 1 and obtain $\mathcal{L}^*$, $\mathbf{w}^*$,
  and $\mathbf{v}^*$.
Unfix $\mathbf{f}$. Solve problem $\mathscr{P}1.1\,(\mathcal{L}^*, \mathbf{w}^*, \mathbf{v}^*)$ and obtain $\mathbf{f}^*$.
Output: $\mathcal{L}^*$, $\mathbf{w}^*$, $\mathbf{v}^*$ and $\mathbf{f}^*$.
```

V. Numerical Results

In this section, we evaluate the performance of the proposed joint optimization algorithms with numerical simulation. We consider 5 - 25 APs randomly deployed with inter-site distance (ISD) of 200 meters, within which UEs are uniformly distributed. Each AP is equipped with 4 transmit antennas, and each UE is endowed with one antenna. A pathloss model of 140 + 36.7log10(d) is assumed for the link distance d between APs and UEs. The utilized noise power σ^2 is -94dBm. A total network bandwidth of 20 MHZ is used. Similar to [24], we set $P_i^{max} = 4W$, $P_i^d = 5.6W$, $P_i^{sleep} = 5.05W$, and $\eta = 0.36$. The required minimum uplink and downlink data rate is 5 Mbps and 10 Mbps, respectively.

For the computation tasks offloaded to the APs, we use $D_k \sim$ Unif $([0, 2d_{avg}])$, where d_{avg} = 1kbits [11]. 330 cycles/byte, typical for Gzip, is adopted for U_k [13]. F_i^{max} is 10 Megacycles/second, and T_{ik}^{max} is 1 second.

Since radio and computation resources are jointly optimized, we consider benchmarks incorporating scenarios involving various selection methods for APs and the number of CPU cycles, including CPU cycles allocation using (24). The $f_{i,k}$ allocation as in (24) is termed *Disjoint Resource Allocation* [26]. For clarity of simulation results, the benchmarks and our proposed joint optimization algorithms are defined below.

- *All-AP active and Disjoint Resource Allocation (ADRA)*: All APs are in operation. Also, each task's $f_{i,k}$ at the AP is assigned using (24), i.e the allocated number of CPU cycles is set proportional to the amount of the computational load of each UE.
- *Optimal-AP number and Disjoint Resource Allocation (ODRA)*: Optimal number of APs are selected for UEs' communication and computation support, while others are put into the sleep-mode. Disjoint Resource Allocation is applied to each task's $f_{i,k}$.
- *Proposed Algorithm*: Joint optimization of the active AP number, the beamforming vectors, and the computation resource allocation.

In Fig. 2, we show the performance of the joint optimization algorithm relative to the other discussed benchmarks as the traffic grows. The ODRA consumes much less power than the ADRA evidently due to the fewer number of operating APs. Considerable power is saved from the switched off RRHs. Due to the efficient transmission power by the optimization of the precoding vectors, the proposed joint optimization saves even more power than the ODRA.

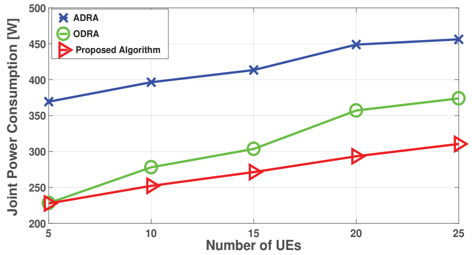

Fig. 2. Total power consumption versus number of UEs

In order to evaluate the energy efficiency performance of our proposed scheme with network densification, we fix the number of UEs at 25 and increase the number of APs. The result is presented in Fig. 3. The ODRA and the joint optimization algorithm yield significant energy savings even with increased AP densification. The impact of optimizing the computational resource is isolated for illustration in Fig. 4. Here, we compare the AP CPU cycles allocation by Disjoint Resource Allocation with the proposed joint optimization. As expected, the ADRA dissipates more computation power than the ODRA despite both schemes use the same CPU cycles allocation method. This is as a result of having a smaller number of APs in the active mode in the ODRA scheme. The computation power of the proposed joint optimization shows more power is conserved in processing offloaded tasks than in the ODRA even though the same number of APs are engaged.

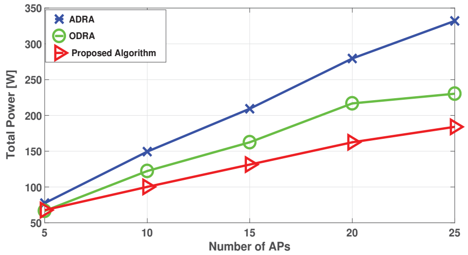

Fig. 3. Total power consumption versus number of APs

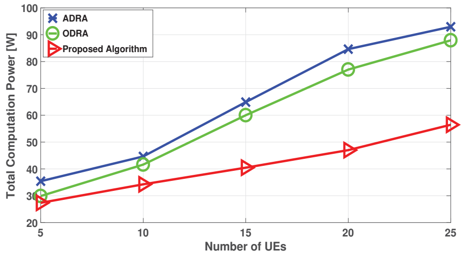

Fig. 4. Total computation power consumption versus number of UEs

We extend the evaluation to the impact of increasing the CPU cycles at the APs on the computation power and the

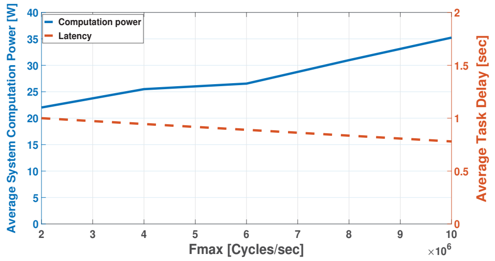

Fig. 5. Impact of maximum AP CPU cycles on computation power consumption and task processing latency

average latency of the computation tasks. This is performed using the proposed joint optimization scheme, and shown in Fig. 5. The number of offloaded tasks and APs are fixed, each at 15. With increased available CPU cycles for tasks processing, the computation completion latency drops. However, with more computation resources used, more power is consumed. Interestingly, the computation power increase and the corresponding decrease in task processing latency are within the specified constraints limits due to joint optimization of the communication and computation resources. For instance, each task latency is within the 1 second deadline regardless of the available AP maximum CPU cycles.

VI. Conclusion

In this paper, we have proposed a joint optimization strategy for achieving energy efficient network in which UEs' computation-intensive tasks are processed at the network edge while satisfying the computation latency requirement. The optimization method entails the selection of active APs. The downlink and uplink beamforming vectors, and CPU capacity allocation are optimized for efficient transmission and low computation latency, respectively. The simulation results show significant power savings by active AP selection and the joint optimization of CPU cycles and the dual link beamforming vectors.

Acknowledgement

This work is partially supported by the US National Science Foundation under Grant No. 1731675 and No. 1810174.

References

[1] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu, and X. Shen, "Synergy of big data and 5g wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 12–18, February 2018.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, 2016.

[3] Y. Chen, X. Wen, Z. Lu, H. Shao, and W. Jing, "Cooperation-enabled energy efficient base station management for dense small cell networks," *Wireless Networks*, vol. 23, no. 5, pp. 1611–1628, 2017.

[4] S.-H. Park, O. Simeone, and S. S. Shitz, "Joint optimization of cloud and edge processing for fog radio access networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7621–7632, 2016.

[5] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. Leung, "Fog radio access networks: Mobility management, interference mitigation and resource optimization," *arXiv preprint arXiv:1707.06892*, 2017.

[6] M. Peng and K. Zhang, "Recent advances in fog radio access networks: Performance analysis and radio resource allocation," *IEEE Access*, vol. 4, pp. 5003–5009, 2016.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[8] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1038–1051, April 2016.

[9] P. He, S. Zhang, L. Zhao, and X. Shen, "Multichannel power allocation for maximizing energy efficiency in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5895–5908, July 2018.

[10] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 3872–3882, August 2013.

[11] Y. Mao, J. Zhang, K. Ben Letaief, Y. Wang, J. T. Y. Kwok, Q. Yao, L. NI, L.-H. Xiong, X. He, J. Xia *et al.*, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," *Biotechnology for Biofuels*, vol. 10, no. 115, 2017.

[12] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.

[13] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, 2017.

[14] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11 255–11 268, 2017.

[15] L. Chen, S. Zhou, and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," *arXiv preprint arXiv:1701.07405*, 2017.

[16] S. Wang, X. Huang, Y. Liu, and R. Yu, "Cachinmobile: An energy-efficient users caching scheme for fog computing," in *Communications in China (ICCC), 2016 IEEE/CIC International Conference on.* IEEE, 2016, pp. 1–6.

[17] S. Luo, R. Zhang, and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in c-ran," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 494–508, 2015.

[18] F. Wang and X. Zhang, "Dynamic interface-selection and resource allocation over heterogeneous mobile edge-computing wireless networks with energy harvesting," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).* IEEE, 2018, pp. 190–195.

[19] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, 2017.

[20] 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.

[21] J. Opadere, Q. Liu, and T. Han, "Energy-efficient rrh sleep mode for virtual radio access networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference.* IEEE, 2017, pp. 1–6.

[22] Q. Liu, T. Han, and G. Wu, "Computing resource aware energy saving scheme for cloud radio access networks," in *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on.* IEEE, 2016, pp. 541–547.

[23] H. Dahrouj and W. Yu, "Coordinated beamforming for the multicell multi-antenna wireless system," *IEEE transactions on wireless communications*, vol. 9, no. 5, 2010.

[24] Y. Shi, J. Zhang, and K. B. Letaief, "Group sparse beamforming for green cloud-ran," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2809–2823, 2014.

[25] T. Baran, D. Wei, and A. V. Oppenheim, "Linear programming algorithms for sparse filter design," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1605–1617, 2010.

[26] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.