Joint Wireless and Computing Resources Allocation in Multi-Cell MEC

Ming Zeng
Department of Electrical and Computer Engineering, Laval University, Quebec, Canada
Email: ming.zeng@gel.ulaval.ca

Abstract—This paper addresses joint wireless and computing resource allocation in mobile edge computing (MEC) systems with several access points and with the possibility that users connect to many access points, and utilize the computation capability of many servers at the same time. The problem of sum transmission energy minimization under response time constraints is considered. It is proved, that the optimization problem is non-convex. The complexity of optimization of a part of the system parameters is investigated, and based on these results an Iterative Resource Allocation procedure is proposed, that converges to a local optimum. The performance of the joint resource allocation is evaluated by comparing it to lower and upper bounds defined by less or more flexible multi-cell MEC architectures. The results show that the free selection of the access point is crucial for good system performance.

I. INTRODUCTION

Mobile edge computing (MEC) represents a promising technology to reduce the latency for 5G and beyond networks [1], [2]. It allows users to offload their computational intensive task to servers in close proximity, and thus, significantly enhancing their computation capacities and prolonging their lifespan. The whole process in general consists of the following three sequential phases [3]: 1) uplink data transmission; 2) task processing at the MEC server; and 3) results feedback through the downlink. The uplink and downlink transmission phases involve the allocation of wireless resources, while the task processing phase concerns the computational resources. As a result, a joint allocation of wireless and computational resources is required to deliver a decent system performance [4].

Initially, MEC was applied to single-user systems, where the optimization variables include the transmission power, the offloading decision and ratio as well as the central processing unit (CPU) frequency for computing [5], [6]. The single-user systems were then extended to the multi-user ones [7]–[9], where how to share the wireless and computational resources among the users directly affect the system performance. Recently, non-orthogonal multiple access is envisioned as a promising access technology [10], and its application to MEC has been considered, for example in [11]. Note that the above mentioned works only apply to single-cell systems. The general scenario of multi-cell systems is attracting great attention recently [12]–[14]. Compared with single-cell systems, a new problem that emerges in multi-cell systems is how to match the users to the access points (APs).

In this paper we consider a multi-AP multi-user MEC system, where each user can access multiple APs and utilize their computation resource. This generalizes the previous works [12]–[14], where each user only offloads to one AP. Meanwhile, as in [14], we consider the general case with flexible bandwidth allocation across both the APs and users. The system objective is to minimize the sum energy consumption under response time constraints. The formulated problem is shown to be non-convex. To handle it, we first investigate the complexity of optimization of a part of the system parameters. On this basis, we propose an iterative resource allocation procedure that converges to local optimum. To evaluate the proposed iterative solution, we compare it with the lower and upper bounds defined by less or more flexible multi-cell MEC architectures. Presented results validate the necessity of free selection of APs. Meanwhile, binary allocation, where all users select the AP with the highest share of their load provides a performance close to parallel processing. This facilitates its application to large systems, where the level of parallel processing is low.

II. SYSTEM MODEL

We consider a multi-cell MEC system that consists of \( K \) users, and \( M \) APs, each equipped with a MEC server. We denote the set of users by \( K = \{1, \cdots, K\} \), and APs by \( M = \{1, \cdots, M\} \). We consider that each user \( i \in K \) generates a computationally intensive and delay sensitive task, which is characterized by three parameters, the size \( L_i \) of the input data, the number \( W_i \) of CPU cycles required to perform the computation, and the completion time constraint \( D_i \). To save energy consumption at the user for processing the tasks, and satisfy the delay constraint, each user offloads its computing task to one or multiple APs for processing. That is, each user \( i \in K \) offloads part of its input data, i.e., \( L_{i,j} \) to AP \( j, j \in M \), satisfying \( L_{i,j} \geq 0 \) and \( \sum_{j=1}^{M} L_{i,j} = L_i \). For simplicity we assume that this parallel processing has no communication or computation overhead [15]. The objective of the considered MEC system is to minimize the energy consumption for data transmission under the delay constraint, by jointly allocating the data to be sent to the APs, the wireless and the computing resources.

A. Wireless resource management

The overall system bandwidth is \( B \) Hz, which should be appropriately shared among the users. We consider flat fading channel and orthogonal access with frequency division multiple access. Denote the corresponding channel gain for
user \( i \) to AP \( j \) by \( h_{i,j} \). Then, the achievable data rate at user \( i \) to AP \( j \) is given by

\[
R_{i,j} = x_{i,j} \log_2 \left( 1 + \frac{P_{i,j} h_{i,j}}{x_{i,j} N_0} \right),
\]

where \( P_{i,j} \) is the corresponding transmission power, while \( x_{i,j} \) denotes the allocated bandwidth, satisfying \( \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{M}} x_{i,j} = B \). Besides, \( N_0 \) is the noise power spectral density coefficient.

Accordingly, the transmission time and the resulting transmission energy consumption are respectively given by

\[
T_{i,j} = \frac{L_{i,j}}{R_{i,j}} \quad \text{and} \quad E_{i,j} = \frac{L_{i,j} P_{i,j}}{R_{i,j}}.
\]

### B. Computing resource management

Let us denote the computational capacity of the MEC server for AP \( j, j \in \mathcal{M} \) by \( C_j \). The energy consumption is minimized when \( T_{i,j} + Q_{i,j} = D_i, \forall i \in \mathcal{K}, j \in \mathcal{M} \) holds and the transmission power is set as

\[
P_{i,j} = \frac{N_0 x_{i,j}}{h_{i,j}} \left( 2 \frac{R_{i,j}^{\min}}{x_{i,j}} - 1 \right), \forall i \in \mathcal{K}, j \in \mathcal{M}
\]

where \( R_{i,j}^{\min} \) is the minimum rate that still fulfills the delay requirement, i.e., \( R_{i,j}^{\min} = \frac{L_{i,j}}{x_{i,j} \log_2 \left( 1 + \frac{P_{i,j} h_{i,j}}{x_{i,j} N_0} \right)} \).

**Proof:** When \( x_{i,j}, q_{i,j} \) and \( L_{i,j} \) are given, the energy consumption of the users is independent, and minimizing the total energy consumption is equivalent to minimizing that of each user on each AP. Without loss of generality, we look at \( E_{i,j} \), which can be reformulated as

\[
E_{i,j} = \frac{L_{i,j} P_{i,j}}{R_{i,j}^{\min} x_{i,j} \log_2 \left( 1 + \frac{P_{i,j} h_{i,j}}{x_{i,j} N_0} \right)}
\]

According to (7), it is clear that \( E_{i,j} \) increases with \( P_{i,j} \). Therefore, \( E_{i,j} \) is minimized when the minimum power is used. Meanwhile, to satisfy the delay constraint, we have \( R_{i,j} = x_{i,j} \log_2 \left( 1 + \frac{P_{i,j} h_{i,j}}{x_{i,j} N_0} \right) \geq R_{i,j}^{\min} \), i.e., \( P_{i,j} \geq (2 R_{i,j}^{\min} / x_{i,j} - 1) N_0 x_{i,j} / h_{i,j} \). At equality the achieved rate is \( R_{i,j}^{\min} \), which in turn results a transmission time of \( T_{i,j} = D_i - Q_{i,j} \). This concludes the proof.

Based on Theorem 1, Problem P1 can be simplified as

\[
P_2: \min_{x,q,L} \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{M}} N_0 x_{i,j} \left( D_i - \frac{\eta L_{i,j}}{q_{i,j}} \right) \left( \frac{x_{i,j} / L_{i,j}}{2 \frac{x_{i,j}}{L_{i,j}} - 1} \right)
\]

s.t.

\[
\sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{M}} x_{i,j} = B \quad \text{(8a)}
\]

\[
\sum_{i \in \mathcal{K}} q_{i,j} = C_j, \forall j \in \mathcal{M} \quad \text{(8b)}
\]

\[
\sum_{i \in \mathcal{K}} L_{i,j} = L_i, \forall i \in \mathcal{K} \quad \text{(8c)}
\]

This is the problem that we will evaluate in detail.

**Theorem 2:** Problem P2, that is, the optimal joint allocation of bandwidth \( x \), computing resource \( q \) and data size \( L \) is a non-convex problem.

**Proof:** P2 requires to jointly allocate bandwidth, computing resource and data size, and is difficult to handle. In the following, we show that P2 is actually non-convex. Without loss of generality, we only consider \( E_{i,j} \). We first consider
$E_{i,j}$ over $x_{i,j}$ and $L_{i,j}$. Denote its Hessian matrix as $H_{i,j}$, which is

$$H_{i,j} = \frac{N_0}{h_i} 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2 \left( \begin{array}{c}
\frac{D_i}{h_i} - \eta \frac{L_{i,j}}{q_{i,j}} x_{i,j} \\
\frac{D_i}{h_i} - \eta \frac{L_{i,j}}{q_{i,j}} x_{i,j} 
\end{array} \right).$$

(9)

It can be verified that $\det(H_{i,j}) \geq 0$ does not always hold. For example, by setting $D_i = 0.1, \eta = 0.5, L_{i,j} = 0.1, x_{i,j} = 0.1, q_{i,j} = 1, N_0 = 1, h_i = 1$, we obtain $\det(H_{i,j}) = -3.1437 \times 10^{13} < 0$. Since all leading principle minors of a convex function should be greater or equal than 0, this indicates $P_2$ is non-convex.

IV. COMPLEXITY OF SUBPROBLEMS AND THE JOINT RESOURCE ALLOCATION ALGORITHM

Problem $P_2$ considers three decision variables, $L$, $q$ and $x$. In this section we provide a discussion about the complexity of the subproblems, where some of the variables are considered as given input parameters. This discussion helps us to find ways for decomposing the optimization problem into tractable subproblems. It also guides future system design, where the network, the computing resources, and the application may be controlled by three different parties.

A. Complexity of Subproblems

Let us first consider subproblems of $P_2$ with a single free decision variable.

Theorem 3: Subproblems of $P_2$, where two of the three variables of $L$, $q$ and $x$ are fixed, are convex problems.

Proof: To prove the theorem, we need to consider the following three subcases:

a) Optimizing data size $L$. According to $H_{i,j}$, we have

$$\frac{\partial^2 E_{i,j}}{\partial L_{i,j}^2} = \frac{N_0 D_i^2}{h_i x_{i,j}} \left( \frac{L_{i,j}}{N_0} / x_{i,j} \right)^2 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2 > 0.$$

Therefore, the considered problem is convex.

b) Optimizing bandwidth $x$. According to $H_{i,j}$, we have

$$\frac{\partial^2 E_{i,j}}{\partial x_{i,j}^2} = \frac{L_{i,j}}{h_i} \left( D_i - \eta \frac{L_{i,j}}{q_{i,j}} x_{i,j} \right)^2 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2 > 0.$$

Therefore, the considered problem is also convex.

c) Optimizing computing resource $q$. We replace variables $q_{i,j}$ with $t_{i,j} = D_i - \eta L_{i,j} / q_{i,j}$ to get

$$\min_{t_{i,j}} \sum_{i \in K} \frac{N_0}{h_i} x_{i,j} t_{i,j} \left( \frac{L_{i,j}}{2^{\frac{L_{i,j}}{N_0} / x_{i,j}}} - 1 \right) \quad (10a)$$

subject to

$$\sum_{i \in K} \frac{\eta L_{i,j}}{D_i - t_{i,j}} = C_j, \forall j \in M$$

(10b)

In $(10)$, equality $(10b)$ is clearly not affine, and thus, the feasible set is non-convex. To address it, we relax the equality constraint and substitute $(10b)$ with

$$\sum_{i \in K} \frac{\eta L_{i,j}}{D_i - t_{i,j}} \leq C_j, \forall j \in M \quad (11)$$

As a consequence of Theorem 1, the energy consumption decreases if $q_{i,j}$ is increased. Thus, for the optimal solution, equality is achieved in $(11)$, which means substituting $(10b)$ with $(11)$ will not change the solution. Then, for inequality constraint $(11)$, its second derivative is $\sum_{i \in K} \frac{2n L_{i,j}}{(D_i - t_{i,j})^2} > 0$, and thus, it is convex. For the objective function, we have

$$\frac{\partial^2 E_{i,j}}{\partial t_{i,j}^2} = \frac{N_0 L_{i,j}^2}{h_{i,j} x_{i,j}} t_{i,j}^2 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2 > 0.$$

Therefore, $(10)$ with the relaxed constraint $(11)$ is convex.

Now let us consider the joint optimization problems, where one of the variables is fixed, while the other two are optimized jointly.

Theorem 4: Subproblems of $P_2$, where $L$ is one of the decision variables, are non-convex problems.

Proof:

a) Optimizing data size $L$ and transmission bandwidth $x$ under fixed processing power allocation $q$. This has been shown in Theorem 3, namely, the proof of the non-convexity of $P_2$.

b) Optimizing data size $L$ and processing power $q$ under fixed transmission bandwidth allocation $x$. In this case, the Hessian matrix is given by

$$\tilde{H}_{i,j} = \frac{N_0 x_{i,j}}{h_{i,j}} \left[ \tilde{H}_{i,j}(1,1) \tilde{H}_{i,j}(1,2) \tilde{H}_{i,j}(2,2) \right],$$

where

$$\tilde{H}_{i,j}(1,1) = \frac{D_i^2}{x_{i,j}^2} \left( \frac{D_i}{h_{i,j}} - \eta L_{i,j} / q_{i,j} \right)^2 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2,$$

$$\tilde{H}_{i,j}(1,2) = \frac{\left( \frac{L_{i,j}}{N_0} / x_{i,j} \right)^2}{\eta L_{i,j}^2 / q_{i,j} \ln 2^2 / x_{i,j}^2},$$

$$\tilde{H}_{i,j}(2,2) = \left(-\frac{2 L_{i,j}}{q_{i,j} \ln 2^2 / x_{i,j}^2} \right)^2 2^{\frac{L_{i,j}}{N_0} / x_{i,j}} \ln 2^2.$$
function (12a). The Hessian matrix is given by (11), which is convex. Last, let us consider the objective of user

\[ L \]

follows two iterative steps: i) the Data Allocation Algorithm and processing resource allocation is convex. This completes the proof.

In this section, an Iterative Resource Allocation algorithm is proposed to solve the non-convex joint resource allocation problem P2. As shown in Algorithm 1, the proposed algorithm follows two iterative steps: i) the Data Allocation Algorithm (DAA) updates \( L \) to allocate the data, for given bandwidth and computing resource allocation \( x \) and \( q \), and ii) the Bandwidth and Computing resource Allocation Algorithm (BCAA) updates \( x \) and \( q \) to allocate bandwidth and computing resource, for given \( L \). We denote by \( E_H^t \) and \( E_F^t \) the energy consumption of user \( i \) after optimizing \( t_i \) and \( x_i \), respectively, and \( \epsilon \) is the stop condition. Additionally, \( L_{i,j}^{(0)} \) denotes the initial value for \( L_{i,j} \), and it can be obtained using a fixed allocation, e.g., equal allocation, or a random allocation, e.g., following a uniform distribution.

**The Data Allocation Algorithm (DAA):** The data allocation problem under given bandwidth and computing resource is shown to be convex in Theorem 3. Therefore, we can use the Karush-Kuhn-Tucker (KKT) condition to derive the optimal \( L_{i,j} \). The KKT condition for data \( L_{i,j} \) is given by (13) at the top of next page. Note that in (13) \( \lambda \) is the Lagrange dual variable.

For given \( \lambda \), the above equation can be used to obtain \( L_{i,j} \). Specifically, we have

\[
\frac{\partial g(L_{i,j})}{\partial L_{i,j}} = \frac{N_0}{h_{i,j}(D_i - \eta L_{i,j}/q_{i,j})^2} \left( 2 \frac{L_{i,j}}{q_{i,j}} - 1 \right) \ln 2 - 1 > 0, \]

which indicates that \( g(L_{i,j}) \) grows with \( L_{i,j} \), and thus a bisection search can be used to obtain \( L_{i,j} \) by comparing \( g(L_{i,j}) \) with 0. Now the problem lies in how to obtain \( \lambda \). When \( \lambda \) is increased, \( L_{i,j} \) will decrease to ensure \( g(L_{i,j}) \). Meanwhile, \( \lambda^* = \sum_{i=1}^{M} L_{i,j} = L_i \) needs to hold. Consequently, \( \lambda \) can also be obtained with bisection search, by comparing \( \lambda^* \) with \( L_{i,j} \).

The resulting DAA consists of two loops: an outer loop to find the value of \( \lambda \) and an inner loop to determine the data allocation \( L_i \). The computational complexity is \( O(KM \log_2(\lambda) \log_2(L_i)) \).

**The Bandwidth and Computing resource Allocation Algorithm (BCAA):** Under given data allocation, the joint bandwidth and computing resource allocation problem is shown to be a convex one in Theorem 3, and thus the optimal solution can be obtained using convex optimization tools, e.g., interior-point method. Nonetheless, considering that the bandwidth allocation is global for the entire network, while the computing resource allocation is local at each AP, we propose to apply an iterative algorithm to solve the joint bandwidth and computing resource allocation. The algorithm is presented in detail in our previous work [14], here we summarize it briefly. The proposed algorithm consists of two iterative steps: i) the Bandwidth Allocation Algorithm (BAA) updates \( x \) to allocate bandwidth across and within the APs, for given \( t \) (\( t_{i,j} = D_i - \eta L_{i,j}/q_{i,j} \)), and ii) the Computation resource Allocation Algorithm (CAA) updates \( t \) to allocate the computing resource at each AP, for given bandwidth allocation \( x \).

**Bandwidth Allocation Algorithm (BAA):** Assume that the computing resource allocation \( t \) is given. Then the KKT condition for \( x_{i,j} \) is given by

\[
f(x_{i,j}) = \frac{N_0 t_{i,j}}{h_{i,j}} \left( 2 \frac{L_{i,j}}{t_{i,j}} - 1 \right) \ln 2 - 1 + \beta = 0,
\]

where \( \beta \) is the introduced auxiliary variable, satisfying \( \beta > 0 \). As for \( L_{i,j} \) in (13), the bisection method can be used to obtain \( \beta \) and \( x_{i,j} \). Likewise, the resulting BAA consists of two loops: an outer loop to find the value of \( \beta \) and an inner loop to determine the bandwidth allocation \( x \). The computational complexity is \( O(KM \log_2(B) \log_2(\beta)) \).
Computing resource Allocation Algorithm (CAA): Under given bandwidth allocation, the computing resource allocation is independent across the APs. Thus, the energy minimization for each AP is equivalent to that of the overall system. Let us consider AP $j$, $j \in \mathcal{M}$. The KKT condition for $t_{i,j}$ is given by

$$
g(L_{i,j}) = \frac{N_0 x_{i,j}}{h_{i,j}} \left[ \frac{D_i \ln 2}{\lambda x_{i,j} (D_i - \eta L_{i,j}/q_{i,j})} - \frac{\eta}{q_{i,j}} \right]^{2} \ln 2 - 1 \right]
+ \frac{W_{i,j}}{(D_i - t_{i,j})^2 \mu_j}
= 0,
$$

where $\mu_j$ is the introduced auxiliary variable, satisfying $\mu_j \geq 0$. As for $L_{i,j}$ and $x_{i,j}$, the bisection method can be used to obtain $\mu_j$ and $t_{i,j}$. Likewise, the resulting CAA consists of two loops: an outer loop to find the value of $\mu_j$ and an inner loop to determine the computing resource allocation $t_{i,j}$. The computational complexity is $O(KM \log_2(D_i) \log_2(\mu_j))$.

BAA and CAA have to be repeated until convergence. Since the bandwidth and computing resource allocation problem is convex, convergence is guaranteed and the converged local optimum is also the global optimum.

**Convergence and Complexity:**

**Theorem 6:** The Iterative Resource Allocation algorithm converges in finite steps.

**Proof:** In both lines 9 and 10 of Algorithm 1, the energy consumption decreases, or remains unchanged. Since there is a lower bound for the energy consumption, e.g., 0, the Iterative Resource Allocation algorithm always terminates, either by reaching the lower bound, or by achieving a decrease less than $\epsilon$. Therefore, convergence is guaranteed. Denote the number of iterations required for Algorithm 1 and BCAA to converge by $I$ and $I_0$, respectively. The total computational complexity is $O(KM \log_2(D_i) \log_2(\lambda) + I_0 \log_2(B) \log_2(\beta) + I_0 \log_2(D_i) \log_2(\mu_j))$.

**V. Numerical Results**

In this section numerical results are presented to evaluate the effectiveness of the proposed energy minimization algorithm. Specifically, first we evaluate the convergence of the Iterative Resource Allocation algorithm. Then we evaluate what is the level of distributed processing under the optimized resource allocation, to see whether this possibility leads to significant performance gains.

We consider a small cell multi-AP multi-user scenario, where image processing tasks are offloaded to the edge computing servers. The network parameters are set as for example in [8, 21, 22], while the task requirements follow [8, 23, 24]. More specifically, the number of APs and users are set to $M = 4$ and $K = 8$. The users are placed uniformly randomly within the entire coverage region. The pathloss model follows $30.6 + 36.7 \log_{10}(d)$, where $d$ is the distance in meter. The total bandwidth is 10 MHz, while the thermal noise density is -174 dBm/Hz. The data size of the computing task is 1.5 Mbits, while the delay constraint is 0.5 s. The computing coefficient $\eta$ is $10^3$, and thus the computing need is 1.5 G CPU cycles. The computing capacity at each AP is 25 G CPU cycles/s.

Let us first investigate how the proposed iterative algorithm, described in Algorithm 1, converges. We consider three different initializations of the user data: Prop-1 denotes the case when the data for each user are equally allocated among the APs; Prop-2 is obtained following a uniformly random data size allocation; and finally Prop-3 represents the case when 90% of the data are allocated to the AP with the best channel condition, while the rest are equally allocated to the remaining APs. The stop condition is $\epsilon = 10^{-2}$ mJ. According to the results, the algorithm converges to the same solution, independently from the initial data allocation. This indicates that the proposed algorithm is robust to the initialization. Moreover, Prop-3 requires the least number of iterations to approach the converged solution in all scenarios, which implies the proposed algorithm converges to a solution where users associate most of the data to their best APs.

We then plot the energy consumption versus the delay requirement in Fig. 2. As expected, the energy consumption decreases as the delay requirement becomes looser, for all considered scenarios. Lastly, we investigate how much the opportunity of parallel processing is utilized in the allocations. Fig. 3 considers only the users that connect to more than one AP, and shows what is the largest share of load sent to one of them, namely $\max_{j} L_{i,j}/L_i$. The values are rather high for all cases, showing that most of the users have a preferred AP. This validates why an unbalanced initialization as Prop-3 on Fig. 1 leads to fast convergence for the Iterative Resource Allocation.
Fig. 2: Energy consumption versus delay under different network scenarios.

Fig. 3: Ratio of largest load share versus delay under different network scenarios.

VI. CONCLUSION
In this paper we investigated the joint wireless and computing resource allocation for a multi-AP multi-user MEC network. The general case with parallel processing and global bandwidth sharing was considered, and the system objective was to minimize the sum transmission energy under response time constraints. The formulated problem was shown to be non-convex. To address it, we first investigated the complexity of optimizing a part of the system parameters, and based on the results proposed an Iterative Resource Allocation procedure with guaranteed convergence. Presented numerical results show that the proposed iterative algorithm converges rapidly to the local optimum. Moreover, by comparing the proposed iterative algorithm with the lower and upper bounds, it is clear that free selection of APs is crucial for obtaining decent system performance.

REFERENCES
[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” IEEE Commun. Surv. Tutor., vol. 19, no. 4, pp. 2322–2358, 2017.
[2] M. Zeng, W. Hao, O. A. Dobre, Z. Ding, and H. V. Poor, “Massive mimo-assisted mobile edge computing: Exciting possibilities for computation offloading,” IEEE Veh. Technol. Mag., vol. 15, no. 2, pp. 31–38, Jun. 2020.
[3] A. Al-Shuwaili and O. Simeone, “Energy-efficient resource allocation for mobile edge computing-based augmented reality applications,” IEEE Wireless Commun. Lett., vol. 6, no. 3, pp. 398–401, Jun. 2017.
[4] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks,” IEEE Signal Process. Mag., vol. 31, no. 6, pp. 45–55, Nov. 2014.
[5] Y. Zhang, et al., “Energy-optimal mobile cloud computing under stochastic wireless channel,” IEEE Trans. Wireless Commun., vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
[6] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling,” IEEE Trans. Commun., vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[7] M. Zeng, L. Jiao, W. Hao, and X. Fu, “Efficient offloading-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[8] C. You, K. Huang, H. Chae, and B. H. Kim, “Energy-efficient resource allocation for mobile edge computing offloading,” IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[9] M. Zeng and V. Fodor, “Energy-efficient resource allocation for noma-assisted mobile edge computing,” in Proc. IEEE PIMRC, Sep. 2018, pp. 1794–1799.
[10] W. Hao et al., “Codebook-based max-min energy-efficient resource allocation for uplink mmwave MIMO-NOMA systems,” IEEE Trans. Commun., vol. 67, no. 12, pp. 8303–8314, Dec. 2019.
[11] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[12] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[13] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[14] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[15] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[16] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[17] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[18] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[19] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[20] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[21] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[22] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[23] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[24] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[25] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[26] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[27] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[28] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[29] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.
[30] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” Ad Hoc Netw., vol. 98, p. 102060, Mar. 2020.