Joint Task Offloading Scheduling and Transmit Power Allocation for Mobile-Edge Computing Systems

Yuyi Mao[†], Jun Zhang[†], and Khaled B. Letaief[†*], *Fellow, IEEE*
[†]Dept. of ECE, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[*]Hamad bin Khalifa University, Doha, Qatar
Email: {ymaoac, eejzhang, eekhaled}@ust.hk

***Abstract*—Mobile-edge computing (MEC) has emerged as a prominent technique to provide mobile services with high computation requirement, by migrating the computation-intensive tasks from the mobile devices to the nearby MEC servers. To reduce the execution latency and device energy consumption, in this paper, we jointly optimize task offloading scheduling and transmit power allocation for MEC systems with multiple independent tasks. A low-complexity sub-optimal algorithm is proposed to minimize the weighted sum of the execution delay and device energy consumption based on alternating minimization. Specifically, given the transmit power allocation, the optimal task offloading scheduling, i.e., to determine the order of offloading, is obtained with the help of flow shop scheduling theory. Besides, the optimal transmit power allocation with a given task offloading scheduling decision will be determined using convex optimization techniques. Simulation results show that task offloading scheduling is more critical when the available radio and computational resources in MEC systems are relatively balanced. In addition, it is shown that the proposed algorithm achieves near-optimal execution delay along with a substantial device energy saving.**

***Index Terms*—Mobile-edge computing, task offloading scheduling, power control, flow shop scheduling, convex optimization.**

I. Introduction

The rapid development of mobile applications with advanced features has brought great pressure on mobile computing systems. However, the limited processing capability of mobile devices becomes an obstacle to fulfill such a requirement. Mobile-edge computing (MEC) has emerged as a promising technique to resolve this issue, which offers computation capability within the radio access network in contrast to conventional cloud computing systems that use remote public clouds [1], [2]. By offloading the computation-intensive tasks from the mobile devices to the nearby MEC servers, the quality of computation experience, including the latency and device energy consumption, could be greatly improved [3].

Nevertheless, the efficiency of an MEC system largely depends on the adopted computation offloading policy, which should be carefully designed by taking the characteristics of the computation tasks and wireless channels into account [5]-[14]. In [5], energy consumption for mobile execution and computation offloading for single-user MEC systems was minimized via dynamic voltage frequency scaling and data transmission scheduling, respectively. A delay-optimal task scheduling algorithm for single-user MEC systems was proposed in [6]. For multi-user MEC systems, a decentralized computation offloading policy was proposed in [7], and joint sub-carrier and CPU time allocation was investigated in [8].

Inspired by the fact that many applications can be divided into a set of dependent sub-tasks, fine-grained computation offloading (a.k.a. code partitioning) has also been widely studied most recently [9]-[14]. Assuming serial implementation of communication and computation, a heuristic offloading algorithm was proposed in [9], while the joint optimization of radio resource and code partitioning was investigated in [10]. Besides, a dynamic code partitioning algorithm was proposed in [11], which is adaptive to the time-varying wireless data rate and the random application requests. In order to reduce the execution latency, parallel implementation of communication and computation was adopted in [12]-[14], where the code partitioning policy design becomes much more challenging. In [12], a code partitioning algorithm was developed based on dynamic programming for applications with a tree topology. This study was extended to applications with general topologies in [13], where the solution has exponential complexity. By leveraging the structures of the application topology, a code partitioning algorithm was proposed to reduce the computation complexity based on message passing in [14]. However, these studies assume that each offloaded sub-task is allocated with certain amount of communication bandwidth and computation resource at the MEC server, i.e., transmitting the input data of multiple tasks (executing multiple tasks at the MEC server) concurrently is allowed. Although such assumption makes the design more tractable, it may be impractical for MEC systems with limited resources, e.g., for MEC servers with a single communication channel and a single-core CPU.

In this paper, we consider a single-user MEC system with multiple independent computation tasks requesting for mobile-edge execution, assuming parallel implementation of task offloading and execution. A radio resource limited system is considered, where each time the input data of only one task can be offloaded, while different tasks are executed by the MEC server sequentially. In this case, the transmission and execution processes of different tasks are coupled, and the task offloading scheduling, i.e., the order of the tasks to be offloaded, becomes a new design dimension. We formulate a joint task offloading scheduling and transmit power allocation problem with the objective of minimizing the weighted sum of the execution delay and device energy consumption, which is a highly non-trivial *mixed integer nonlinear programming*

This work is supported by the Hong Kong Research Grants Council under Grant No. 16200214.

978-1-5090-4183-1/17/$31.00 ©2017 IEEE

(MINLP) problem. A low-complexity sub-optimal algorithm is proposed based on alternating minimization. In particular, the optimal task offloading scheduling with a given transmit power allocation is obtained with the aid of *flow shop scheduling theory*, while the optimal transmit power allocation with a given task offloading scheduling decision is determined based on *convex optimization*. Simulation results shall show that task offloading scheduling is more critical when the radio and computational resources are relatively balanced. Besides, near-optimal execution delay along with a substantial device energy saving can be achieved by the proposed algorithm.

II. System Model

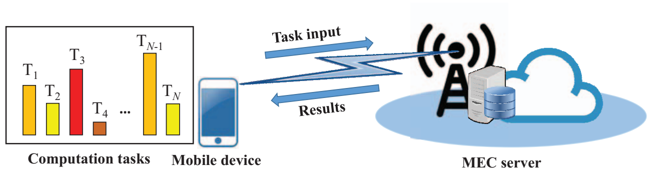

Fig. 1. A mobile-edge computing system with a mobile device and an MEC server.

We consider a single-user mobile-edge computing (MEC) system as shown in Fig. 1, which consists of a mobile device and an MEC server. The MEC server is a small-scale data center deployed by the telecom operator and communicates with the mobile device through the wireless channel. A cloud-clone associated with the mobile device is running at the MEC server. Hence, the MEC server can execute the computation tasks on behalf of the mobile device [4]. Besides, the distance between the mobile device and the MEC server is denoted as L, and the system bandwidth is represented by ω.

A. Computation Task Model

We assume that the mobile device has N independent computation tasks that need to be executed, where the set of tasks is denoted as $\mathcal{T} \triangleq \{\mathrm{T}_1, \cdots, \mathrm{T}_N\}$. Each computation task is characterized by a two-tuple of parameters, $\langle d_i, c_i \rangle$, where d_i (in bits) is the amount of task input data, and c_i (in CPU cycles/bit) is the workload. The values of d_i and c_i depend on the nature of the computation tasks and can be obtained through off-line measurements [15]. Denote $\mathbf{d} = [d_1, \cdots, d_N]$ and $\mathbf{c} = [c_1, \cdots, c_N]$. In this paper, we focus on the scenarios that the mobile device has very limited computational resource, and hence all the computation tasks should be offloaded to the MEC server for mobile-edge execution [16]. The scenarios that the mobile device has more powerful computation capability will be handled in our future work.

B. Task Offloading and Mobile-Edge Execution Model

In order to offload the computation tasks for mobile-edge execution, the task input data should be transmitted to the MEC server. We denote the task offloading scheduling decision for the N tasks as $\boldsymbol{\sigma} \triangleq [\sigma_1, \sigma_2, \cdots, \sigma_N]$, which is a permutation of the task indices and T_{σ_i} is the ith task being offloaded to the MEC server. Hence, $\boldsymbol{\sigma}$ satisfies $\sigma_i \in \{1, \cdots, N\}$ and $\sigma_i \neq \sigma_j, j \neq i, \forall i, j = 1, \cdots, N$. The mobile device is equipped with a single antenna and it can transmit the input data of one computation task at each time. Thus, the transmission rate for task T_i can be expressed as

$$R(p_i) = \omega \log_2 \left(1 + \frac{g_0 (L_0/L)^\theta p_i}{N_0 \omega}\right), \tag{1}$$

where p_i is the transmit power for task T_i, g_0 is the path-loss constant, θ is the path-loss exponent, L_0 is the reference distance, and N_0 is the noise power spectral density at the receiver of the MEC server.

The MEC server has a single-core CPU and executes the offloaded tasks one after another. A task buffer is employed at the MEC server to store the tasks that have been offloaded but not yet executed, which is assumed to be sufficiently large for simplicity. Besides, it executes the computation task with a *first-come-first-serve (FCFS)* fashion. Thus, the execution order of the tasks is the same as the task offloading order. The CPU-cycle frequency at the MEC server is denoted as f_{ser} (in Hz). Besides, the computation results are assumed to be of small size and thus the feedback delay is ignored.

The execution delay and device energy consumption are two critical design considerations in MEC systems, and both of them depend on the adopted task offloading scheduling and transmit power allocation policy. In the next section, we will formulate an optimization problem to minimize the weighted sum of the execution delay and device energy consumption, by jointly designing the offloading scheduling and transmit power allocation for the computation tasks.

III. Problem Formulation

In this section, we will first analyze the execution delay and device energy consumption with given task offloading scheduling decision $\boldsymbol{\sigma}$ and transmit power allocation vector $\mathbf{p} = [p_1, \cdots, p_N]$. The joint task offloading scheduling and transmit power allocation problem will then be formulated.

The mobile-edge execution for a computation task cannot be started until the following two conditions are satisfied: First, the task input data is ready at the MEC server. Second, the CPU at the MEC server is available for executing a new task. Denote the ready time of the task input data and the completion time for the jth task being offloaded (i.e., T_{σ_j}) as $t^j_{\text{ready}}(\boldsymbol{\sigma}, \mathbf{p})$ and $t^j_{\text{comp}}(\boldsymbol{\sigma}, \mathbf{p})$, respectively. Thus, $t^j_{\text{ready}}(\boldsymbol{\sigma}, \mathbf{p})$ is given by

$$t^j_{\text{ready}}(\boldsymbol{\sigma}, \mathbf{p}) = \sum_{k \le j} \frac{d_{\sigma_k}}{R(p_{\sigma_k})}, j = 1, \cdots, N, \tag{2}$$

which is the sum transmission time of tasks $\mathrm{T}_{\sigma_1}, \cdots, \mathrm{T}_{\sigma_j}$.

The completion time of task T_{σ_j} not only depends on the ready time of its task input data, but also couples with the completion time of the task being offloaded previously, i.e., $\mathrm{T}_{\sigma_{j-1}}$. Consequently, $t^j_{\text{comp}}(\boldsymbol{\sigma}, \mathbf{p})$ can be expressed as the following recursive form:

$$t^j_{\text{comp}}(\boldsymbol{\sigma}, \mathbf{p}) = \begin{cases} t^j_{\text{ready}}(\boldsymbol{\sigma}, \mathbf{p}) + d_{\sigma_j} c_{\sigma_j} f^{-1}_{\text{ser}}, & j = 1 \\ \max\{t^j_{\text{ready}}(\boldsymbol{\sigma}, \mathbf{p}), t^{j-1}_{\text{comp}}(\boldsymbol{\sigma}, \mathbf{p})\} & \\ \qquad + d_{\sigma_j} c_{\sigma_j} f^{-1}_{\text{ser}}, & j > 1, \end{cases} \tag{3}$$

where $d_{\sigma_j} c_{\sigma_j} f_{\text{ser}}^{-1}$ is the mobile-edge execution time of task T_{σ_j}, and $t_{\text{comp}}^{N}(\boldsymbol{\sigma}, \mathbf{p})$ is the execution delay for $\mathcal{T}$.

The device energy consumption for offloading the N computation tasks is simply the transmit energy consumption, which is independent with the task offloading scheduling decision and can be written as

$$E_{\text{tr}}(\mathbf{p}) = \sum_{i=1}^{N} p_i \cdot \frac{d_i}{R(p_i)} = \sum_{j=1}^{N} p_{\sigma_j} \cdot \frac{d_{\sigma_j}}{R\left(p_{\sigma_j}\right)}. \qquad (4)$$

Consequently, the joint task offloading scheduling and transmit power allocation problem can be formulated as

$$\mathcal{P}_1 : \min_{\boldsymbol{\sigma}, \mathbf{p}} \; t_{\text{comp}}^{N}(\boldsymbol{\sigma}, \mathbf{p}) + \eta \cdot E_{\text{tr}}(\mathbf{p}) \qquad (5)$$

$$\text{s.t.} \;\; \sigma_j \in \{1, \cdots, N\}, \sigma_i \neq \sigma_j, j \neq i, \forall i, j \qquad (6)$$

$$0 \leq p_i \leq p_{\max}, i = 1, \cdots, N. \qquad (7)$$

The objective function in $\mathcal{P}_1$ is the weighted sum of the execution delay and device energy consumption with η (in $\sec \cdot \mathrm{J}^{-1}$) as the weighting factor, which is able to adjust the tradeoff between the execution delay and device energy consumption performance. (6) guarantees the task offloading scheduling decision is valid, and (7) stands for the peak and non-negative transmit power constraints for the mobile device.

It can be easily noticed that $\mathcal{P}_1$ is an MINLP problem as both the integer variables $\boldsymbol{\sigma}$ and the continuous variables $\mathbf{p}$ need to be optimized, which is very challenging to solve. In principle, the optimal solution for $\mathcal{P}_1$ can be obtained via exhaustive search, which, however, is practically infeasible due to high complexity. For instance, there will be $20! \approx 2.43 \times 10^{18}$ different permutations for $N = 20$, and for each permutation, the optimal transmit power allocation needs to be determined. As a result, instead of finding the optimal solution, we will propose a low-complexity sub-optimal algorithm for $\mathcal{P}_1$ based on alternating minimization in the next section.

IV. Sub-optimal Joint Task Offloading Scheduling and Transmit Power Allocation

In this section, we will propose a low-complexity sub-optimal algorithm for $\mathcal{P}_1$ based on the idea of optimizing the task offloading scheduling and the transmit power allocation alternately.

A. Optimal Task Offloading Scheduling

We first develop an optimal task offloading scheduling algorithm for a given transmit power allocation vector by invoking techniques from *flow shop scheduling theory* [17]. To begin with, we introduce a notation for a type of two-machine flow shop scheduling problems in the following definition.

Definition 1: $F2|(perm),(pmtn)|C_{\max}$ denotes the type of flow shop scheduling problems that satisfy the following properties:

a) There are N independent jobs and each of them should be processed sequentially by two machines M_1 and M_2.
b) All jobs are available for M_1 from the beginning, and they will be available for M_2 immediately after the completion of the processing at M_1.
c) Each machine can only process one job at each time.
d) Permutation scheduling is considered, i.e., the jobs go through the two machines in the same order, which is encoded as $(perm)$ in the notation.
e) Non-preemptive processing is enforced, i.e., interruption is not allowed once the processing of a job starts, which is encoded as $(pmtn)$ in the notation.
f) Each machine has a buffer with infinite capacity to store the arrived but not yet processed jobs.
g) The goal is to minimize the makespan of the N jobs, which is encoded as $C_{\max}$ in the notation.

We find that the task offloading scheduling problem is a two-machine flow shop scheduling problem, as demonstrated in the follow lemma.

Lemma 1: For a given transmit power allocation vector $\mathbf{p}$, $\mathcal{P}_1$ is an $F2|(perm),(pmtn)|C_{\max}$ type of flow shop scheduling problem.

Proof: In the task offloading scheduling problem, each computation task needs to go through two procedures before its completion, which are the transmission of the task input data and the mobile-edge execution at the MEC server. Hence, the transmitter of the mobile device and the MEC server can be regarded as M_1 and M_2 in Definition 1, respectively. Thus, the job processing time in the two machines correspond to the transmission time and mobile-edge execution time, respectively. By analogy, it can be checked that the properties in Definition 1 also holds for the task offloading scheduling problem. As a result, $\mathcal{P}_1$ with a given $\mathbf{p}$ is an $F2|(perm),(pmtn)|C_{\max}$ type of flow shop scheduling problem. ■

Since the task offloading scheduling problem is an $F2|(perm),(pmtn)|C_{\max}$ type of flow shop scheduling problem, the *Johnson's Algorithm* can be applied to find the optimal solution [17]. In the Johnson's Algorithm, the task set $\mathcal{T}$ is partitioned into two disjoint subsets $\mathcal{F}$ and $\mathcal{G}$, where $\mathcal{F} \triangleq \{\mathrm{T}_i \in \mathcal{T} | \frac{d_i}{R(p_i)} < \frac{d_i c_i}{f_{\text{ser}}}\}$ and $\mathcal{G} \triangleq \{\mathrm{T}_i \in \mathcal{T} | \frac{d_i}{R(p_i)} \geq \frac{d_i c_i}{f_{\text{ser}}}\}$. The tasks in set $\mathcal{F}$ will be scheduled before those in $\mathcal{G}$ according to the ascending order of the value of $d_i / R(p_i), \mathrm{T}_i \in \mathcal{F}$, and the tasks in set $\mathcal{G}$ will be scheduled according to the descending order of $d_i c_i / f_{\text{ser}}, \mathrm{T}_i \in \mathcal{G}$. Details of the Johnson's Algorithm are summarized in Algorithm 1, where the computation overhead mainly comes from the sorting procedures in Line 4. Hence, if the *Quicksort Algorithm* is used as the sorting algorithm, the complexity of Algorithm 1 is $\mathcal{O}(N \log N)$ [18], i.e., the task offloading scheduling problem can be solved optimally in polynomial time.

B. Optimal Transmit Power Allocation

In this subsection, we investigate the optimal transmit power allocation for the computation tasks with a given task offloading scheduling decision $\boldsymbol{\sigma}$. Note that both the execution delay and the device energy consumption depend on the transmit power allocation vector $\mathbf{p}$, and $t_{\text{comp}}^{N}(\boldsymbol{\sigma}, \mathbf{p})$ is given in a recursive form, while the objective function in $\mathcal{P}_1$ given $\boldsymbol{\sigma}$ is non-differentiable, which makes it difficult to solve. To overcome this issue, we introduce a set of auxiliary variables $\tilde{\mathbf{t}}_{\boldsymbol{\sigma}} \triangleq \left[\tilde{t}_{\sigma_1}, \cdots, \tilde{t}_{\sigma_N}\right]$ and formulate a modified version of the

Algorithm 1 The Johnson's Algorithm For The Optimal Task Offloading Scheduling Decision

```
Input: $\mathcal{T}$, $\mathbf{d}$, $\mathbf{c}$, $\mathbf{p}$, and $f_{\mathrm{ser}}$.
Output: $\boldsymbol{\sigma}^{\mathrm{opt}}$.
Obtain set $\mathcal{F}$ and $\mathcal{G}$.
Sort the computation tasks in set $\mathcal{F}$ and set $\mathcal{G}$ according to the ascending order of the transmission time and the descending order of the execution time at the MEC server, respectively, i.e., $d_{[1]}/R\left(p_{[1]}\right) \leq \cdots \leq d_{[|\mathcal{F}|]}/R\left(p_{[|\mathcal{F}|]}\right), \mathrm{T}_{[i]} \in \mathcal{F}, i=1,\cdots,|\mathcal{F}|$ and $d_{\langle 1\rangle}c_{\langle 1\rangle}f_{\mathrm{ser}}^{-1} \geq \cdots \geq d_{\langle|\mathcal{G}|\rangle}c_{\langle|\mathcal{G}|\rangle}f_{\mathrm{ser}}^{-1}, \mathrm{T}_{\langle j\rangle} \in \mathcal{G}, j=1,\cdots,|\mathcal{G}|$.
$\boldsymbol{\sigma}^{\mathrm{opt}} = [[1],\cdots,[|\mathcal{F}|],\langle 1\rangle,\cdots,\langle|\mathcal{G}|\rangle]$.
```

transmit power allocation problem as

$$\mathcal{P}_2: \min_{\tilde{\mathbf{t}}_{\boldsymbol{\sigma}},\mathbf{p}} \ \tilde{t}_{\sigma_N} + \eta \cdot \sum_{i=1}^{N} p_{\sigma_i} \frac{d_{\sigma_i}}{R\left(p_{\sigma_i}\right)} \tag{8}$$

$$\text{s.t.} \quad (7)$$

$$\tilde{t}_{\sigma_i} \geq \sum_{j=1}^{i} \frac{d_{\sigma_j}}{R\left(p_{\sigma_j}\right)} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, i=1,\cdots,N \tag{9}$$

$$\tilde{t}_{\sigma_i} \geq \tilde{t}_{\sigma_{i-1}} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, i=2,\cdots,N, \tag{10}$$

which is a relaxed version of $\mathcal{P}_1$ given $\boldsymbol{\sigma}$ as $\tilde{t}_{\sigma_i} \geq \max\{\sum_{j=1}^{i} \frac{d_{\sigma_j}}{R(p_{\sigma_j})}, \tilde{t}_{\sigma_{i-1}}\} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, i \geq 2$ and $\tilde{t}_{\sigma_1} \geq \frac{d_{\sigma_1}}{R(p_{\sigma_1})} + \frac{d_{\sigma_1}c_{\sigma_1}}{f_{\mathrm{ser}}}$. In the following lemma, we show such a relaxation is tight.

Lemma 2: If $\langle \mathbf{p}^{\mathrm{opt}}, \tilde{\mathbf{t}}_{\boldsymbol{\sigma}}^{\mathrm{opt}}\rangle$ is an optimal solution for $\mathcal{P}_2$, then $\mathbf{p}^{\mathrm{opt}}$ is also optimal for $\mathcal{P}_1$ with a given task offloading scheduling decision $\boldsymbol{\sigma}$, and $\tilde{t}_{\sigma_N}^{\mathrm{opt}}$ is the corresponding execution delay.

Proof: With the optimal transmit power allocation vector $\mathbf{p}^{\mathrm{opt}}$, we can construct a new feasible $\tilde{\mathbf{t}}'_{\boldsymbol{\sigma}}$ for $\mathcal{P}_2$ as

$$\tilde{t}'_{\sigma_i} = \begin{cases} \frac{d_{\sigma_i}}{R\left(p_{\sigma_i}^{\mathrm{opt}}\right)} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, & i=1 \\ \max\{\tilde{t}'_{\sigma_{i-1}}, \sum_{j=1}^{i} \frac{d_{\sigma_j}}{R\left(p_{\sigma_j}^{\mathrm{opt}}\right)}\} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, & i>1, \end{cases} \tag{11}$$

where $\tilde{t}'_{\sigma_N}$ is no larger than any of the feasible $\tilde{t}_{\sigma_N}$'s for $\mathcal{P}_2$ given $\mathbf{p}^{\mathrm{opt}}$, i.e., $\tilde{t}_{\sigma_N}^{\mathrm{opt}} = \tilde{t}'_{\sigma_N}$ and $\langle \mathbf{p}^{\mathrm{opt}}, \tilde{\mathbf{t}}'_{\boldsymbol{\sigma}}\rangle$ is also optimal for $\mathcal{P}_2$. Therefore, by applying $\mathbf{p}^{\mathrm{opt}}$ to $\mathcal{P}_1$ given $\boldsymbol{\sigma}$, the value of the objective function $\mathrm{Val}_{\mathcal{P}_1}$ equals $\mathrm{Val}_{\mathcal{P}_2}^{\mathrm{opt}}$, where $\mathrm{Val}_{\mathcal{P}_2}^{\mathrm{opt}}$ is the optimal value for $\mathcal{P}_2$. Since $\mathcal{P}_2$ is a relaxation of $\mathcal{P}_1$ given the task offloading scheduling decision, i.e., $\mathrm{Val}_{\mathcal{P}_1}^{\mathrm{opt}} \geq \mathrm{Val}_{\mathcal{P}_2}^{\mathrm{opt}}$ with $\mathrm{Val}_{\mathcal{P}_1}^{\mathrm{opt}}$ as the optimal value for $\mathcal{P}_1$ given $\boldsymbol{\sigma}$, we have $\mathrm{Val}_{\mathcal{P}_1} = \mathrm{Val}_{\mathcal{P}_1}^{\mathrm{opt}}$, i.e., $\mathbf{p}^{\mathrm{opt}}$ is optimal for $\mathcal{P}_1$ given $\boldsymbol{\sigma}$. ■

Thus, we may concentrate on $\mathcal{P}_2$ in order to obtain the optimal transmit power allocation vector. Nevertheless, $\mathcal{P}_2$ is still difficult to solve due to its non-convexity, which is shown in the following lemma.

Lemma 3: $\mathcal{P}_2$ is a non-convex optimization problem.

Proof: The proof can be obtained by verifying the concavity of $\frac{p}{R(p)}$ $(p>0)$, which is omitted for brevity. ■

Fortunately, we are able to transform $\mathcal{P}_2$ into a convex optimization problem with the *change-of-variable* technique similar to the one used in [10]. In particular, by introducing a set of new variables $\xi_{\sigma_i} = 1/R\left(p_{\sigma_i}\right), i=1,\cdots,N$, $\mathcal{P}_2$ can be rephrased as

$$\mathcal{P}_3: \min_{\tilde{\mathbf{t}}_{\boldsymbol{\sigma}},\boldsymbol{\xi}} \ \tilde{t}_{\sigma_N} + C \cdot \sum_{i=1}^{N} d_{\sigma_i}\xi_{\sigma_i}\left(2^{\frac{1}{\omega\xi_{\sigma_i}}} - 1\right) \tag{12}$$

$$\text{s.t.} \quad (10)$$

$$\xi_{\sigma_i} \geq D, i=1,\cdots,N \tag{13}$$

$$\tilde{t}_{\sigma_i} \geq \sum_{j=1}^{i} d_{\sigma_j}\xi_{\sigma_j} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}}, i=1,\cdots,N, \tag{14}$$

where $C \triangleq \frac{\eta N_0 \omega}{g_0(L_0/L)^{\theta}}$ and $D \triangleq \frac{1}{\omega \log_2\left(1+\frac{g_0(L_0/L)^{\theta}}{N_0\omega}p_{\max}\right)}$. In the following lemma, we show that $\mathcal{P}_3$ is a convex problem.

Lemma 4: $\mathcal{P}_3$ is a convex optimization problem.

Proof: Denote $\psi(\xi) \triangleq \xi\left(2^{\frac{1}{\omega\xi}} - 1\right)$. The second-order derivative of $\psi(\xi)$ is given by $\frac{d^2\psi(\xi)}{d\xi^2} = 2^{\frac{1}{\omega\xi}}\frac{\ln^2 2}{\omega^2\xi^3} \geq 0, \forall\xi > 0$, i.e., $\psi(\xi)$ is a convex function of ξ $(\xi > 0)$. Since the objective function is a summation of convex and linear functions, and the constraints are linear, $\mathcal{P}_3$ is a convex optimization problem [19]. ■

Hence, the optimal transmit power allocation vector can be obtained numerically using convex optimization solvers such as CVX [20]. Interestingly, we find that the optimal transmit power allocation vector, i.e., $p_{\sigma_1}^{\mathrm{opt}},\cdots,p_{\sigma_N}^{\mathrm{opt}}$, is a non-increasing sequence, as demonstrated in the following corollary.

Corollary 1: Suppose $1 \leq i < j \leq N$, then $p_{\sigma_i}^{\mathrm{opt}} \geq p_{\sigma_j}^{\mathrm{opt}}$, where $\mathbf{p}^{\mathrm{opt}}$ is the optimal power allocation vector for $\mathcal{P}_2$ (also for $\mathcal{P}_1$ given $\boldsymbol{\sigma}$).

Proof: Let $\langle \boldsymbol{\xi}^{\mathrm{opt}}, \tilde{\mathbf{t}}_{\boldsymbol{\sigma}}^{\mathrm{opt}}\rangle$ be the optimal solution for $\mathcal{P}_3$. Since $\xi_{\sigma_i} = 1/R\left(p_{\sigma_i}\right)$, it is equivalent to show $\xi_{\sigma_i}^{\mathrm{opt}} \leq \xi_{\sigma_j}^{\mathrm{opt}}, 1 \leq i < j \leq N$. We first write the partial Lagrangian for $\mathcal{P}_3$ as

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\xi}, \tilde{\mathbf{t}}_{\boldsymbol{\sigma}}, \boldsymbol{\lambda}_{\boldsymbol{\sigma}}\right) &= \tilde{t}_{\sigma_N} + C \cdot \sum_{i=1}^{N} d_{\sigma_i}\xi_{\sigma_i}\left(2^{\frac{1}{\omega\xi_{\sigma_i}}} - 1\right) \\ &+ \sum_{i=1}^{N} \alpha_{\sigma_i}\left(\sum_{j=1}^{i} d_{\sigma_j}\xi_{\sigma_j} + \frac{d_{\sigma_i}c_{\sigma_i}}{f_{\mathrm{ser}}} - \tilde{t}_{\sigma_i}\right) \\ &+ \sum_{k=2}^{N} \beta_{\sigma_k}\left(\tilde{t}_{\sigma_{k-1}} - \tilde{t}_{\sigma_k} + \frac{d_{\sigma_k}c_{\sigma_k}}{f_{\mathrm{ser}}}\right), \end{aligned} \tag{15}$$

where $\boldsymbol{\alpha}_{\boldsymbol{\sigma}} = [\alpha_{\sigma_1},\cdots,\alpha_{\sigma_N}] \succeq \mathbf{0}$ and $\boldsymbol{\beta}_{\boldsymbol{\sigma}} = [\beta_{\sigma_2},\cdots,\beta_{\sigma_N}] \succeq \mathbf{0}$ are the Lagrangian multipliers for constraints (14) and (10), respectively. Define $\boldsymbol{\lambda}_{\boldsymbol{\sigma}} \triangleq [\boldsymbol{\alpha}_{\boldsymbol{\sigma}}, \boldsymbol{\beta}_{\boldsymbol{\sigma}}]$. Since $\mathcal{P}_3$ is a convex problem, we consider its dual problem $\max_{\boldsymbol{\lambda}_{\boldsymbol{\sigma}}\succeq\mathbf{0}} g\left(\boldsymbol{\lambda}_{\boldsymbol{\sigma}}\right)$, where $g\left(\boldsymbol{\lambda}_{\boldsymbol{\sigma}}\right) = \inf_{\tilde{t}_{\sigma_i},\xi_{\sigma_i}\geq D,\forall i} \mathcal{L}\left(\boldsymbol{\xi}, \tilde{\mathbf{t}}_{\boldsymbol{\sigma}}, \boldsymbol{\lambda}_{\boldsymbol{\sigma}}\right)$. If the optimal dual variables are given by $\boldsymbol{\lambda}_{\boldsymbol{\sigma}}^{\mathrm{opt}}$, the optimal primal variables $\boldsymbol{\xi}^{\mathrm{opt}}$ can be obtained by solving

$$\min_{\xi_{\sigma_i}\geq D,\forall i} C\cdot\sum_{i=1}^{N} d_{\sigma_i}\xi_{\sigma_i}\left(2^{\frac{1}{\omega\xi_{\sigma_i}}} - 1\right) + \sum_{i=1}^{N} \alpha_{\sigma_i}^{\mathrm{opt}}\left(\sum_{j=1}^{i} d_{\sigma_j}\xi_{\sigma_j}\right), \tag{16}$$

for which the optimal solution is given by $\xi_{\sigma_i}^{\star}\left(\boldsymbol{\lambda}_{\boldsymbol{\sigma}}^{\text{opt}}\right) = \max\{D, \varphi\left(\sum_{j=i}^{N} \alpha_{\sigma_j}^{\text{opt}}\right)\}$. Here, when $x = 0$, $\varphi(x) \triangleq +\infty$, and when $x > 0$, $\varphi(x)$ denotes the root of equation $-d\psi(\xi_{\sigma_i})/d\xi_{\sigma_i} = 1 - 2^{\frac{1}{\omega\xi_{\sigma_i}}}\left(1 - \frac{\ln 2}{\omega\xi_{\sigma_i}}\right) = x/C$. Since $d\psi(\xi_{\sigma_i})/d\xi_{\sigma_i}$ is an increasing function of ξ_{σ_i} ($\xi_{\sigma_i} > 0$), $\lim_{\xi_{\sigma_i}\to 0} d\psi(\xi_{\sigma_i})/d\xi_{\sigma_i} = -\infty$ and $\lim_{\xi_{\sigma_i}\to +\infty} d\psi(\xi_{\sigma_i})/d\xi_{\sigma_i} = 0$, we can show that for $i < j$, $\xi_{\sigma_i}^{\text{opt}} \leq \xi_{\sigma_j}^{\text{opt}}$ as $\alpha_{\sigma_l}^{\text{opt}} \geq 0, \forall l = 1, \cdots, N$, which completes the proof. ■

An intuitive explanation for Corollary 1 is that the completion time of T_{σ_i} will affect those of all the subsequent tasks, i.e., $\mathrm{T}_{\sigma_{i+1}}, \cdots, \mathrm{T}_{\sigma_N}$. Thus, it is desirable to allocate a higher transmit power in order to reduce its transmission time.

C. The Alternating Minimization Algorithm

In the proposed algorithm, the task offloading scheduling decision and the transmit power allocation vector will be updated in an alternating manner, for which, the key steps are summarized in Algorithm 2. Since the Johnson's Algorithm determines the optimal task offloading scheduling with a given transmit power allocation vector, and the solution for $\mathcal{P}_3$ offers the optimal transmit power allocation with a given task offloading scheduling decision, the value of the objective function in $\mathcal{P}_1$ decreases after each update, i.e., the convergence of Algorithm 2 is guaranteed.

Algorithm 2 Sub-Optimal Joint Task Offloading Scheduling and Transmit Power Allocation Algorithm

```
Input: $\mathbf{p}^{\text{prop}} = p_{\max}\mathbf{1}$, $\boldsymbol{\sigma}^{\text{prop}} = [1, \cdots, N]$, $\text{iter}_{\max} = 50$, $\text{Val}_{\text{new}} = t_{\text{comp}}^{N}(\boldsymbol{\sigma}^{\text{prop}}, \mathbf{p}^{\text{prop}}) + \eta \cdot E_{\text{tr}}(\mathbf{p}^{\text{prop}})$, $\text{Val}_{\text{old}} = \text{Val}_{\text{new}} + 10$, $\epsilon = 10^{-7}$, and $I = 0$.
Output: $\mathbf{p}^{\text{prop}}$ and $\boldsymbol{\sigma}^{\text{prop}}$.
While $\text{Val}_{\text{old}} - \text{Val}_{\text{new}} \geq \epsilon$ and $I \leq \text{iter}_{\max}$ do
    Set $I = I + 1$.
    Set $\text{Val}_{\text{old}} = \text{Val}_{\text{new}}$.
    Update $\boldsymbol{\sigma}^{\text{prop}}$ using Algorithm 1 with the transmit power allocation vector $\mathbf{p}^{\text{prop}}$.
    Update $\mathbf{p}^{\text{prop}}$ with the task offloading scheduling decision $\boldsymbol{\sigma}^{\text{prop}}$ by solving $\mathcal{P}_3$.
    Set $\text{Val}_{\text{new}} = t_{\text{comp}}^{N}(\boldsymbol{\sigma}^{\text{prop}}, \mathbf{p}^{\text{prop}}) + \eta \cdot E_{\text{tr}}(\mathbf{p}^{\text{prop}})$.
End while
```

V. Simulation Results

In this section, we first investigate the impact of the optimal task offloading scheduling on the delay performance. The proposed joint task offloading scheduling and transmit power allocation algorithm will then be evaluated. In simulations, the task input data size and the task workload are assumed to be uniformly distributed, i.e., $d_i \sim \text{Unif}([0, 2d_{\text{avg}}])$ and $c_i \sim \text{Unif}([0, 2c_{\text{avg}}])$ where $d_{\text{avg}} = 1$ kbits and $c_{\text{avg}} = 797.5$ cycles/bit [15]. The CPU speed at the MEC server is set to be $f_{\text{ser}} = 1$ GHz unless otherwise specified.

A. Impact of The Optimal Task Offloading Scheduling

We investigate the impact of the optimal task offloading scheduling on the execution delay by setting $\eta = 0\ \text{sec} \cdot \text{J}^{-1}$, which is obtained by Algorithm 1. A random scheduling algorithm is adopted as the performance benchmark, where the task offloading scheduling decision $\boldsymbol{\sigma}$ is a random permutation of the task indices. Three scenarios with $R = \frac{f_{\text{ser}}}{c_{\text{avg}}} \approx 1.25$ Mbps, $R = \frac{2f_{\text{ser}}}{c_{\text{avg}}} \approx 2.51$ Mbps, and $R = \frac{f_{\text{ser}}}{2c_{\text{avg}}} \approx 0.63$ Mbps, are considered, where the average transmission time of the task input data equals the average processing time at the MEC server in the first case.

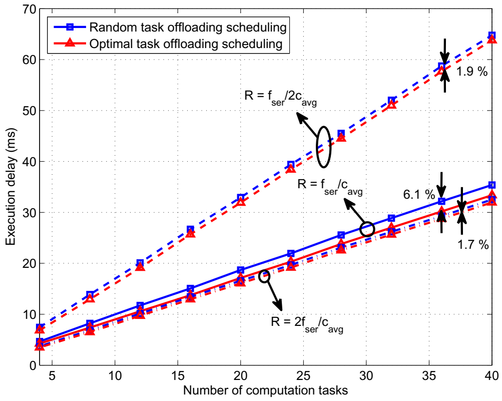

Fig. 2. Execution delay vs. the number of computation tasks.

In Fig. 2, we show the relationship between the execution delay and the number of computation tasks. It can be seen that the execution delay increases linearly with the number of tasks, and a larger transmission rate results in a lower execution delay, which agree with intuitions. However, increasing the transmission rate from 0.63 to 1.25 Mbps brings a significant delay reduction, while the delay reduction by further doubling the transmission rate shrinks. This is because in the latter case, the system bottleneck turns into the limited computational resource at the MEC server. Besides, the proposed optimal task offloading scheduling algorithm outperforms the random task offloading scheduling for all cases, while the performance gain achieved by the optimal algorithm is most obvious when $R = \frac{f_{\text{ser}}}{c_{\text{avg}}}$ (e.g. 6.1% for $N = 35$). In other words, the optimal task offloading scheduling is more critical when the available radio resource and computational resource are relatively balanced, i.e., neither of them dominates the other.

B. Joint Task Offloading Scheduling and Transmit Power Allocation

We evaluate the proposed joint task offloading scheduling and transmit power allocation algorithm by setting $g_0 = -40$ dB, $L_0 = 1$ m, $L = 100$ m, $\theta = 4$, $\omega = 1$ MHz, $N_0 = -174$ dBm/Hz, $p_{\max} = 100$ mW, and $N = 20$.

By varying η, the values of the objective function in $\mathcal{P}_1$ under the proposed algorithm and a benchmark algorithm with random task offloading scheduling and maximum transmit power are shown in Fig. 3. From the curves, we see a significant performance improvement achieved by the proposed algorithm. Besides, the performance gain becomes more dramatic as η increases since the transmit power allocation vector used in the benchmark scheme is non-adaptive to η.

The relationship between the device energy consumption and the execution delay is shown in Fig. 4, from which, we see

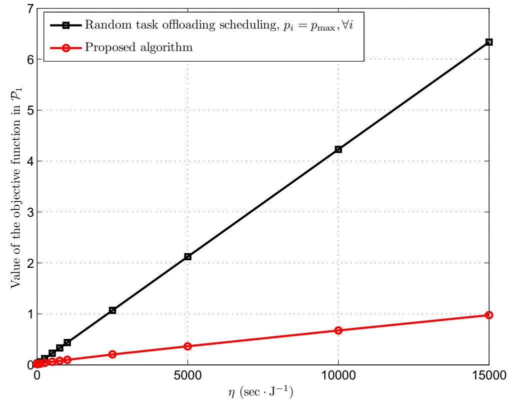

Fig. 3. Value of the objective function in $\mathcal{P}_1$ vs. η.

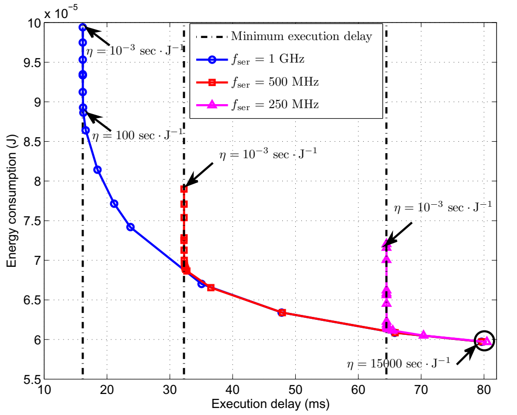

Fig. 4. Device energy consumption vs. execution delay. (The device energy consumption with $p_i = p_{\max}$ is 4.21×10^{-4} J.)

the device energy consumption decreases while the execution delay increases with η. It is also shown that allocating the transmission energy beyond a threshold has no effect on improving the delay performance. This indicates that a large device energy consumption reduction can be achieved without loss of the optimal delay performance under the proposed algorithm with a suitable choice of η. For instance, with $\eta = 100$ sec $\cdot$ J^{-1}, 78% of the device energy consumption is saved for $f_{\text{ser}} = 1$ GHz. This is due to the fact that the delay performance is limited by the computational resource, similar to what was observed in Fig. 2. In addition, when the device energy consumption becomes sufficiently small (η becomes large enough), the curves for different CPU speeds at the MEC sever converge, i.e., the delay performance is constrained by the radio resource. This reveals a fundamental design principle for MEC systems: Once the system is constrained by the available radio resource, there is no need to deploy too much computational resources.

VI. Conclusions

In this paper, we investigated joint task offloading scheduling and transmit power allocation for MEC systems with multiple independent tasks. Based on flow shop scheduling and convex optimization, we proposed a low-complexity sub-optimal algorithm to minimize the weighted sum of the execution delay and device energy consumption. It was found that the optimal task offloading scheduling achieves the most noticeable delay performance improvement when the available radio and computational resources are relatively balanced. Also, near-optimal delay performance together with a large device energy consumption reduction can be achieved by the proposed algorithm. For future investigation, it would be interesting to extend this work for mobile devices with certain computation capability, where the task offloading decision, i.e., whether to offload a task or not, the task offloading scheduling decision, as well as the transmit power allocation need to be jointly optimized.

References

[1] European Telecommunications Standards Institute (ETSI), "Mobile-edge computing-Introductory technical white paper," Sep. 2014.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," [Online.] Available: https://arxiv.org/pdf/1701.01090v1.pdf

[3] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129-140, Feb. 2013.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14-23, Oct. 2009.

[5] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569-4581, Sep. 2013.

[6] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Barcelona, Spain, Jul. 2016.

[7] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974-983, Apr. 2015.

[8] Y. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and CPU time allocation for mobile edge computing," in Proc. *IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, Dec. 2016.

[9] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, Canada, Apr. 2014.

[10] P. D. Lorenzo, S. Barbarossa, and S. Sardellitti "Joint optimization of radio resources and code partitioning in mobile edge computing," [Online]. Available: http://arxiv.org/pdf/1307.3835v3.pdf

[11] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991-1995, Jun. 2012.

[12] Y. H. Kao and B. Krishnamachari, "Optimizing mobile computational offloading with delay constraints," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, Dec. 2014.

[13] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comput.*, to appear.

[14] S. Khalili and O. Simeone, "Inter-layer per-mobile optimization of cloud mobile computing: A message passing approach," [Online] Available: http://arxiv.org/pdf/1509.01596v1.pdf

[15] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, Jun. 2010.

[16] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, to appear.

[17] H. Emmons and G. Vairaktarakis, *Flow shop scheduling: Theoretical results, algorithms, and applications.* New York, USA : Springer, 2013.

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithm.* Cambridge, MA: The MIT Press, 2009.

[19] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge, UK: Cambridge University Press, 2004.

[20] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, version 2.0 (beta)," 2013.