Energy-Efficient Resource Allocation for Multi-User Mobile Edge Computing

Junfeng Guo, Zhaozhe Song, Ying Cui
Shanghai Jiao Tong University, China

Zhi Liu
Shizuoka University, Japan

Yusheng Ji
National Institute of Informatics, Japan

***Abstract*— Designing mobile edge computing (MEC) systems by jointly optimizing communication and computation resources, which can help increase mobile batteries' lifetime and improve quality of experience for computation-intensive and latency-sensitive applications, has received significant interest. In this paper, we consider energy-efficient resource allocation schemes for a multi-user mobile edge computing system with inelastic computation tasks and non-negligible task execution durations. First, we establish a mathematical model to characterize the offloading of a computation task from a mobile to the base station (BS) equipped with MEC servers. This computation-offloading model consists of three stages, i.e., task uploading, task executing, and computation result downloading, and allows parallel transmissions and executions for different tasks. Then, we formulate the weighted sum energy consumption minimization problem to optimally allocate the task operation sequence, the uploading and downloading time durations as well as the starting times for uploading, executing and downloading, which is a challenging mixed discrete-continuous optimization problem and is NP-hard in general. We propose a method to obtain an optimal solution and develop a low-complexity algorithm to obtain a sub-optimal solution, by connecting the optimization problem to a three-stage flow-shop scheduling problem and utilizing Johnson's algorithm as well as convex optimization techniques. Finally, numerical results show that the proposed sub-optimal solution outperforms existing comparison schemes.**

***Index Terms*— Mobile edge computing, computation offloading, resource allocation, optimization, flow-shop scheduling.**

I. Introduction

With the support of on-device cameras and embedded sensors, new applications with advanced features, e.g., navigation, augmented reality, interactive online gaming and multi-media transformation like Speech2Text, are emerging. These applications are both computation-intensive and latency-sensitive. Mobile edge computing (MEC) is a promising technology providing an IT service environment and cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network and in close proximity to mobile users to improve quality of experience. In an MEC system, computation tasks of mobile users are uploaded to the base station (BS) and executed at the attached MEC servers. Then, the computation results are transmitted back to the mobiles. With the drastic demands of applications with crucial computation and latency requirements, finite battery lifetime and limited communication and computation resources pose challenges for designing future energy-efficient MEC systems [1].

Designing energy-efficient MEC systems requires the joint allocation of communication and computation resources among distributed mobiles and MEC servers. Emerging research toward this direction considers optimal resource allocation for various types of multi-task MEC systems [2]–[10]. For example, [2]–[4] consider a single-user MEC system with one BS and multiple elastic tasks, and minimize the execution delay of all tasks under the transmission power constraints. References [5]–[8], [10] study a multi-user MEC system with one BS and one inelastic task for each user, and minimize the energy consumption under a hard deadline constraint for each task. In [9], the authors investigate a multi-user MEC system with one BS and multiple independent elastic tasks for each user, and consider the minimization of the overall cost of energy, computation, and delay for all users. In particular, the offloading scheduling [2], [3], [6]–[9], task operation sequence selection [4], [6], storage resource allocation [10], and transmission time (or power) allocation [2]–[10] are considered in the optimizations.

Note that [2]–[6], [8] assume that the size of the computation result of each task is negligible, and fail to take account of the resource consumption for the BS to transmit computation results to mobiles. In addition, [3], [8], [10] assume that task execution durations are negligible. Although [2], [4]–[7], [9] consider non-negligible task execution durations, they ignore the fact that the execution of one task can be conducted during the transmission of another task (i.e, transmissions and executions for different tasks can be conducted in parallel). Thus, [2], [3], [5], [7]–[10] do not consider the optimization of the task operation sequence, which significantly affects the opportunities for parallel processing and hence leads to larger delay under power constraints or larger energy consumption under deadline constraints. Therefore, the obtained solutions in [2]–[10] are not suitable for the applications involving computation-intensive and latency-sensitive tasks with computation results of large sizes, such as augmented reality, interactive online gaming and multi-media transformation.

In this paper, we shall address the above issues. We consider energy-efficient resource allocation schemes for a multi-user MEC system with one BS of computing capability and multiple users, with inelastic computation tasks of non-negligible

Junfeng Guo is also with Science and Technology on Communication Networks Laboratory. The work of Y. Cui was supported by NSFC grant 61401272 and grant 61521062, National 863 project 2015AA01A710 as well as Science and Technology on Communication Networks Laboratory Foundation. The work of Z. Liu was supported by JSPS Kakenhi Grant Numbers 16H02817 and 15K21599. The work of Y.Ji was supported in part by JSPS Kakenhi Project JP16H02817.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

task execution durations. We adopt a more comprehensive task model, specifying each task using three parameters, i.e., the size of the task before execution, workload and size of the computation result. Based on this task model, we first establish a mathematical model to characterize the offloading of a task from a mobile to the BS. This task-offloading model consists of three stages, i.e., task uploading, task executing and computation result downloading, and allows parallel transmissions and executions for different tasks. Then, we formulate the weighted sum energy consumption minimization problem to optimally allocate the task operation sequence as well as the uploading and downloading time durations. The problem is a challenging mixed discrete-continuous optimization problem and is NP-hard in general. We propose a method to obtain an optimal solution and develop a low-complexity algorithm to obtain a sub-optimal solution. Specifically, for given uploading and downloading durations, we connect the problem to a three-stage flow-shop scheduling problem [11] and utilize Johnson's algorithm to obtain an optimal task operation order and the minimum total completion time under a mild condition. For a given total uploading and downloading duration, we obtain the optimal uploading and downloading durations for each task, using convex optimization techniques. Finally, numerical results show that the proposed sub-optimal solution has low-complexity, close-to-optimal performance and outperforms existing comparison schemes.

II. System Model

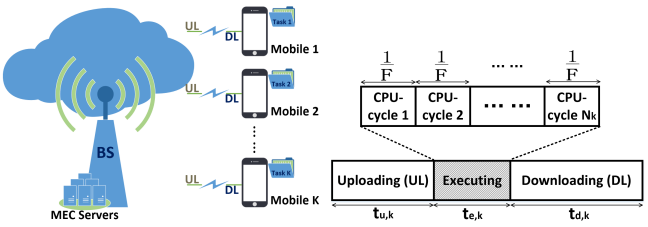
As illustrated in Fig. 1, we consider a multi-user MEC system consisting of one single-antenna BS and K single-antenna mobiles, denoted by set $\mathcal{K} \triangleq \{1, 2, ..., K\}$. The BS has powerful computing capability by running servers of a constant CPU-cycle frequency (in number of CPU-cycles per second) at the network edge. Without loss of generality, we assume that each mobile has one computation-intensive and latency-sensitive (computation) task with deadline T (in seconds), which is offloaded to the BS for executing. Note that multiple tasks with the same deadline can be treated as one super task, whose required computational capability (workload) is the sum of the multiple tasks' workloads.[1]

A. Task Model

We first adopt a more comprehensive computation task model. The computation task at mobile $k \in \mathcal{K}$, referred to as task k, is characterized by three parameters, i.e., the size of the (uploaded) task before computation $L_{u,k} > 0$ (in bits), workload $N_k > 0$ (in number of CPU-cycles), and size of the (downloaded) computation result $L_{d,k} > 0$ (in bits). The computation of each task k has to be accomplished within T seconds.

Remark 1 (Task Model): Note that prior work [2], [3], [5] typically assumes computation results to be negligible in size and hence trivial to download, which may not hold for many applications with computation results of large sizes, such as augmented reality and multi-media transformation. The adopted task model in this paper properly addresses this limitation. In addition, note that the three parameters of a computation task are determined by the nature of the task itself, and for some computation tasks, such as html2text and x264 video encoding, these three parameters can be estimated to certain extent based on some prior offline measurements (please see the measurement table in [13] for an example).

[1]We assume that all tasks have to be executed at the BS due to crucial computation and latency requirements. The optimization results obtained in this paper can be extended to study a more general task scenario, where some tasks can be executed locally and different tasks may have different deadlines [12].



(a) Multi-user MEC system. (b) Three stages.

Fig. 1: System model.

B. Computation-Offloading Model with Non-negligible Task Execution Durations

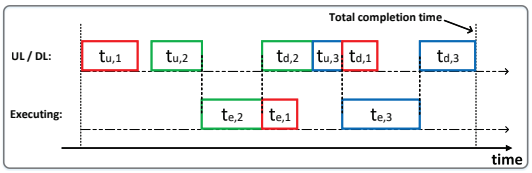
Offloading task k to the BS for executing comprises three sequential stages: 1) uploading task k of $L_{u,k}$ bits from mobile k to the BS; 2) executing task k at the BS (which requires N_k BS CPU-cycles); 3) downloading the computation result of $L_{d,k}$ bits from the BS to mobile k. Let $t_{u,k}$, $t_{e,k}$ and $t_{d,k}$ denote the uploading, execution and downloading durations (in seconds) in the three stages, respectively, where

$$t_{u,k} \geq 0, \ k \in \mathcal{K}, \tag{1}$$

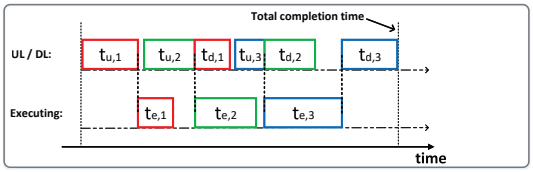
$$t_{e,k} = \frac{N_k}{F}, \ k \in \mathcal{K}, \tag{2}$$

$$t_{d,k} \geq 0, \ k \in \mathcal{K}. \tag{3}$$

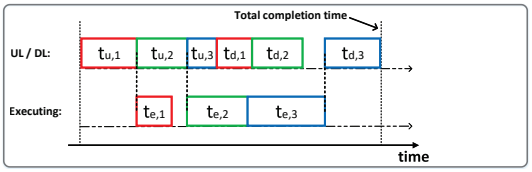
Here, F denotes the fixed CPU-cycle frequency at the BS. Note that $t_{e,k}$ is fixed, while $t_{u,k}$ and $t_{d,k}$ can be optimized. Denote $\mathbf{t}_u \triangleq (t_{u,k})_{k\in\mathcal{K}}$, $\mathbf{t}_e \triangleq (t_{e,k})_{k\in\mathcal{K}}$ and $\mathbf{t}_d \triangleq (t_{d,k})_{k\in\mathcal{K}}$. Although F is usually large, the BS has to handle multiple computation-intensive and latency-sensitive tasks and the overall execution duration of all these offloading tasks may not be negligible in practice. To reflect the impact of the overall execution duration of all tasks, we capture the execution duration of each task in the computation-offloading model. We consider Time Division Multiple Access (TDMA) with Time-Division Duplexing (TDD) operation for transmission, and any uploading duration and downloading duration do not overlap. The execution of one task and the transmission of another task can be conducted at the same time. These make the MEC system more efficient but the computation-offloading model with non-negligible task execution durations is much more complex than the ideal one with negligible task execution durations, as illustrated in Fig. 2(a). In the following,



(a) Arbitrary operation sequences and starting times for uploading, execution and downloading operations.



(b) Same operation sequence for uploading, execution and downloading operation.



(c) Completing the uploading operations of all tasks before starting the downloading operation of any task.

Fig. 2: Illustration example of three operations of all tasks at the BS ($K = 3$). For each task, the duration of each operation is represented by the length of the corresponding rectangle.

we introduce new notations and constraints to mathematically specify this model.

Let $s_{u,k}$, $s_{e,k}$ and $s_{d,k}$ denote the starting times for uploading, executing and downloading task k, respectively. Let $c_{u,k}$, $c_{e,k}$ and $c_{d,k}$ denote the completion times for uploading, executing and downloading task k, respectively. As each of the three stages cannot be interrupted, we first have the following constraints:

$$\begin{cases} s_{u,k} + t_{u,k} = c_{u,k} \\ s_{e,k} + t_{e,k} = c_{e,k} \\ s_{d,k} + t_{d,k} = c_{d,k} \end{cases}, \ k \in \mathcal{K}. \tag{4}$$

To ensure that the uploading, execution and downloading operations of task k are conducted sequentially, we require:

$$\begin{cases} s_{u,k} \geq 0 \\ s_{e,k} \geq c_{u,k} \\ s_{d,k} \geq c_{e,k} \end{cases}, \ k \in \mathcal{K}. \tag{5}$$

By (4) and (5), we know that the total completion time for processing all K tasks is given by $\max_{k\in\mathcal{K}}\{c_{d,k}\}$ [14]. To guarantee the deadline constraint, we have:

$$\max_{k\in\mathcal{K}} \{c_{d,k}\} \leq T. \tag{6}$$

We can have three sequences (orders) for uploading, execution, and downloading of the K tasks, respectively. Following the proof of Lemma 3 in [14], we can show that the three sequences can be made the same without increasing the total completion time, as illustrated in Fig. 2(b). Thus, without loss of generality, we consider the same sequence for uploading, executing and downloading the K tasks, denoted by $\mathbf{S} \in \mathbb{S}$, where $\mathbb{S}$ denotes the set of the $K!$ different permutations of all tasks in $\mathcal{K}$. We let subscript $[k]$ denote the task index at position k of sequence $\mathbf{S}$. From Lemma 1 in [15], we know that, completing the uploading operations of all K tasks before starting the downloading operation of any task will not increase the total completion time, as illustrated in Fig. 2(c). To ensure that at any time, there are at most one task being executed and at most one task being transmitted, we have the following constraints:

$$\begin{cases} s_{u,[k]} \geq c_{u,[k-1]} \\ s_{e,[k]} \geq c_{e,[k-1]} \\ s_{d,[k]} \geq c_{d,[k-1]} \end{cases}, \ k = 2, 3, ..., K, \tag{7}$$

$$s_{d,[1]} \geq c_{u,[K]}. \tag{8}$$

Remark 2 (Non-negligible Task Execution Durations): Note that task execution durations are not considered in [3], [8], and durations for downloading computation results are not captured in [2]–[6], [8], resulting in much simpler task operation models in [2]–[6], [8]. The proposed computation-offloading model in this paper successfully addresses these limitations. Specifically, under this model, the uploading or downloading of a task can be conducted in parallel with the execution of another task, and the task operation sequence greatly affects the total completion time (and hence the energy consumption which will be seen in Section II-C).

C. Energy Consumption Model

Similar to [5] and [16], we consider low CPU voltage at the BS, and model the energy consumption for executing a task as follows.[2] At the BS, the amount of energy consumption for computation in a single CPU-cycle with frequency F is μF^2, where μ is a constant factor determined by the switched capacitance at the MEC servers [5]. Then, the energy consumption for executing task k at the BS is given by:

$$E_{e,k} \triangleq \mu N_k F^2. \tag{9}$$

We now introduce the energy consumption model for task uploading and downloading operations. We consider the block fading channel model. Let h_k denote the channel power gain for mobile k which is assumed to be constant during the T seconds [2]–[9]. Let p_k denote the transmission power of mobile k for uploading task k. Then, the achievable transmission rate (in bit/s) for uploading task k is given by:

$$r_k = B \log_2 \left(1 + \frac{p_k |h_k|^2}{n_0}\right), \tag{10}$$

where B and n_0 are the bandwidth (in Hz) and the complex additive white Gaussian channel noise, respectively. On the other hand, the transmission rate for uploading task k is fixed as $r_k = L_{u,k}/t_{u,k}$, since this is the most energy-efficient

[2]The circuit power is omitted here for simplicity but can be accounted for by adding a constant [5], [16].

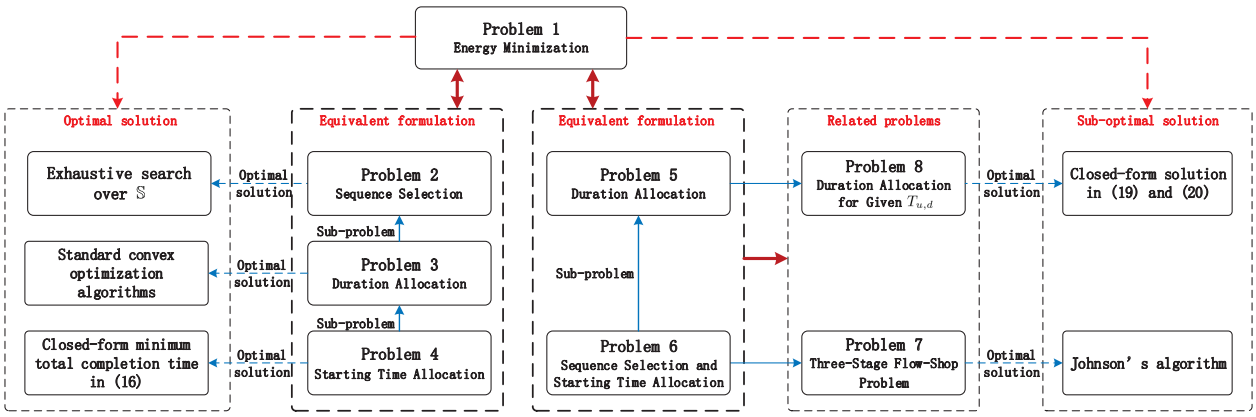


Fig. 3: Proposed optimal and sub-optimal solutions to Problem 1.

transmission method for transmitting $L_{u,k}$ bits in $t_{u,k}$ seconds (due to the fact that $p_k = \frac{n_0}{|h_k|^2}\left(2^{\frac{r_k}{B}}-1\right)$ is a convex function of r_k) [5]. Define $g(x) \triangleq n_0\left(2^{\frac{x}{B}}-1\right)$. Then, we have $p_k = \frac{1}{|h_k|^2} g\left(\frac{L_{u,k}}{t_{u,k}}\right)$. Thus, the energy consumption at mobile k for uploading task k to the BS is given by:

$$E_{u,k}(t_{u,k}) \triangleq p_k t_{u,k} = \frac{t_{u,k}}{|h_k|^2} g\left(\frac{L_{u,k}}{t_{u,k}}\right). \tag{11}$$

Similarly, the energy consumption at the BS for transmitting the computation result of task k to mobile k is given by:

$$E_{d,k}(t_{d,k}) \triangleq \frac{t_{d,k}}{|h_k|^2} g\left(\frac{L_{d,k}}{t_{d,k}}\right). \tag{12}$$

Thus, the energy consumption at the BS for executing task k and transmitting the computation result of task k is given by:

$$E_{\mathrm{BS},k}(t_{d,k}) = E_{e,k} + E_{d,k}(t_{d,k}). \tag{13}$$

The weighted sum energy consumption corresponding to task k is given by:

$$E_k(t_{u,k}, t_{d,k}) = E_{u,k}(t_{u,k}) + \beta E_{\mathrm{BS},k}(t_{d,k}), \tag{14}$$

where $\beta \geq 0$ is the corresponding weight factor. Therefore, the weighted sum energy consumption is given by:

$$E(\mathbf{t}_u, \mathbf{t}_d) = \sum_{k\in\mathcal{K}} E_k(t_{u,k}, t_{d,k}). \tag{15}$$

Note that the weighted sum energy consumption is a convex function of the uploading and downloading time durations for all K tasks.

III. Problem Formulation and Optimal Solution

In this section, we first formulate the energy minimization problem for the multi-user MEC system with non-negligible task execution durations. Then, we propose a method to obtain an optimal solution.

We would like to minimize the weighted sum energy consumption for the multi-user MEC with non-negligible task execution durations under the uploading and downloading duration allocation constraints. Specifically, we have the following optimization problem.

Problem 1 (Energy Minimization):

$$\begin{aligned} E^* \triangleq \min_{\mathbf{S}\in\mathbb{S},\mathbf{s}_u,\mathbf{s}_e,\mathbf{s}_d,\mathbf{t}_u,\mathbf{t}_d} \quad & E(\mathbf{t}_u,\mathbf{t}_d) \\ s.t. \quad & (1),(3),(4),(5),(6),(7),(8), \end{aligned}$$

where $\mathbf{s}_u \triangleq (s_{u,k})_{k\in\mathcal{K}}$, $\mathbf{s}_e \triangleq (s_{e,k})_{k\in\mathcal{K}}$ and $\mathbf{s}_d \triangleq (s_{d,k})_{k\in\mathcal{K}}$.

Problem 1 is a mixed discrete-continuous optimization problem with three types of variables, i.e., the task operation sequence (discrete variable), uploading and downloading durations (continuous variables), as well as starting times for uploading, execution and downloading operations (continuous variables). It is NP-hard in general, which will be seen in the discussion in Section IV. By exploiting structural properties of Problem 1, we propose an equivalent formulation, as shown in Fig. 3. This formulation separates the three types of variables and facilitates the optimization.

Problem 2 (Sequence Selection):

$$\begin{aligned} E^* = \min_{\mathbf{S}} \quad & E^*_{\mathrm{seq}}(\mathbf{S}) \\ s.t. \quad & \mathbf{S}\in\mathbb{S}. \end{aligned}$$

Let $\mathbf{S}^*$ denote the optimal solution. $E^*_{\mathrm{seq}}(\mathbf{S})$ is given by the following sub-problem.

Problem 3 (Duration Allocation for Given $\mathbf{S}$): For all $\mathbf{S}\in\mathbb{S}$, we have

$$\begin{aligned} E^*_{\mathrm{seq}}(\mathbf{S}) \triangleq \min_{\mathbf{t}_u,\mathbf{t}_d} \quad & E(\mathbf{t}_u,\mathbf{t}_d) \\ s.t. \quad & T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d) \leq T, \\ & (1),(3). \end{aligned}$$

Let $(\mathbf{t}^*_u(\mathbf{S}), \mathbf{t}^*_d(\mathbf{S}))$ denote the optimal solution. $T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d)$ is given by the following sub-problem.

Problem 4 (Starting Time Allocation for Given $\mathbf{S},\mathbf{t}_u,\mathbf{t}_d$): For given $\mathbf{S}\in\mathbb{S}$, $\mathbf{t}_u$ and $\mathbf{t}_d$, the minimum total completion time is given by:

$$\begin{aligned} T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d) \triangleq \min_{\mathbf{s}_u,\mathbf{s}_e,\mathbf{s}_d} \quad & c_{d,[K]} \\ s.t. \quad & (4),(5),(7),(8). \end{aligned}$$

$$T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d) = \max\left\{\max_{1\le i\le j\le K}\left\{\left(\sum_{k=1}^{j}t_{e,[k]}-\sum_{k=1}^{j-1}t_{d,[k]}\right)+\left(\sum_{k=1}^{i}t_{u,[k]}-\sum_{k=1}^{i-1}t_{e,[k]}\right)\right\},\sum_{k=1}^{K}t_{u,[k]}\right\}+\sum_{k=1}^{K}t_{d,[k]}. \quad (16)$$

$$\widetilde{T}_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d) = \max_{1\le i\le j\le K}\left\{\left(\sum_{k=1}^{j}t_{e,[k]}-\sum_{k=1}^{j-1}t_{d,[k]}\right)+\left(\sum_{k=1}^{i}t_{u,[k]}-\sum_{k=1}^{i-1}t_{e,[k]}\right)\right\}+\sum_{k=1}^{K}t_{d,[k]}. \quad (17)$$

We can show the following result.[3]

Theorem 1: (Relationship between Problem 1 and Problems 2-4): Problem 1 and Problems 2-4 are equivalent.

Based on Theorem 1, we propose a method to obtain an optimal solution to Problem 1, by solving Problems 2-4. This method is illustrated in Fig. 3. First, we solve Problem 4 for given $\mathbf{S}\in\mathbb{S}$, $\mathbf{t}_u$ and $\mathbf{t}_d$.

Lemma 1: (Minimum Total Completion Time): For given $\mathbf{S}\in\mathbb{S}$, $\mathbf{t}_u$ and $\mathbf{t}_d$, the minimum total completion time is given by (16) (at the top of the this page).

Then, substituting $T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d)$ in (16) into Problem 3, we have a convex optimization problem with $2K$ variables, as $T_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d)$ in (16) is convex (note that it is a positive weighted sum of convex piecewise linear functions). This problem can be solved using standard convex optimization techniques. Finally, we can solve Problem 2 by evaluating all possible choices for $\mathbf{S}\in\mathbb{S}$ using exhaustive search.

IV. Sub-optimal Solution

Note that obtaining an optimal solution to Problem 2 requires solving Problem 3 and Problem 4 up to $K!$ times. The complexity is not acceptable when K is large. In this section, we first propose an equivalent formulation for Problem 1 related to a standard three-stage flow-shop scheduling problem and a convex optimization problem. Then, we develop a low-complexity algorithm to obtain a sub-optimal solution by utilizing Johnson's algorithm [14] and convex optimization techniques.

First, we propose an equivalent formulation for Problem 1, i.e., Problem 5 and Problem 6, as illustrated in Fig. 3. Different from Problems 2-4, this equivalent formulation divides the three types of variables into two groups. One includes the uploading and downloading durations, and the other includes the task operation sequence and the starting times for uploading, execution and downloading operations.

Problem 5 (Duration Allocation):

$$\begin{aligned} E^* = \min_{\mathbf{t}_u,\mathbf{t}_d}\ & E(\mathbf{t}_u,\mathbf{t}_d) \\ s.t.\ & T_F^*(\mathbf{t}_u,\mathbf{t}_d)\le T, \\ & (1),(3). \end{aligned}$$

Let $(\mathbf{t}_u^*,\mathbf{t}_d^*)$ denote the optimal solution. $T_F^*(\mathbf{t}_u,\mathbf{t}_d)$ is the optimal value of the following sub-problem.

[3]We omit all the proofs due to page limitation. Please refer to [12] for details.

Problem 6: (Sequence Selection and Starting Time Allocation): For any $\mathbf{t}_u$ and $\mathbf{t}_d$, we have

$$\begin{aligned} T_F^*(\mathbf{t}_u,\mathbf{t}_d) \triangleq \min_{\mathbf{S}\in\mathbb{S},\mathbf{s}_u,\mathbf{s}_e,\mathbf{s}_d}\ & c_{d,[K]} \\ s.t.\quad & (4),(5),(7),(8). \end{aligned}$$

In the following, we discuss Problem 6 and Problem 5, respectively, based on which we propose a low-complexity sub-optimal solution, as illustrated in Fig. 3. Before interpreting Problem 6, we introduce some background on M-stage flow-shop scheduling problems. In an M-stage flow-shop scheduling problem, all tasks have to be processed on M machines following the same machine order. Each task requires certain fixed processing time on a machine. The objective is to find an operation sequence for processing all tasks on each machine so that a given criterion is optimal. The most commonly studied criterion in literature is the total completion time. When $M\ge 3$, an M-stage flow-shop scheduling problem is NP-hard in general [14]. When $M=3$, the three sequences for processing the tasks on the three machines can be set to be the same without losing optimality, and the optimal sequence can be efficiently obtained by Johnson's algorithm in a special case [14].

By treating $\mathbf{t}_u$, $\mathbf{t}_e$ (given in (2)) and $\mathbf{t}_d$ as the processing times for three separate machines (i.e., uploading machine, executing machine and downloading machine), Problem 6 can be regarded as a three-stage flow-shop scheduling problem with an additional constraint in (8) (i.e., the uploading machine and the downloading machine cannot operate at the same time). By relaxing the additional constraint in (8) and using the expression of the minimum total completion time (without the additional constraint in (8)), we can transform Problem 6 into a standard three-stage flow-shop scheduling problem [14].

Problem 7 (Three-Stage Flow-Shop Scheduling Problem): For any $\mathbf{t}_u$ and $\mathbf{t}_d$, we have

$$\widetilde{T}_F^*(\mathbf{t}_u,\mathbf{t}_d) \triangleq \min_{\mathbf{S}\in\mathbb{S}}\ \widetilde{T}_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d),$$

where the minimum total completion time (without the additional constrain in (8)) $\widetilde{T}_F(\mathbf{S},\mathbf{t}_u,\mathbf{t}_d)$ is given by (17) (at the top of this page), and is obtained by optimizing $\mathbf{s}_u,\mathbf{s}_e,\mathbf{s}_d$ under the constraints in (4), (5), (7). Let $\mathbf{S}^*(\mathbf{t}_u,\mathbf{t}_d)$ denote an optimal solution.

It can be easily verified that Problem 7 is a standard three-stage flow-shop scheduling problem [11], [14]. We now establish the relationship between Problem 6 and Problem 7.

Lemma 2: (Relationship between Problem 6 and Problem 7): Given $\mathbf{t}_u$ and $\mathbf{t}_d$, an optimal solution $\mathbf{S}^*(\mathbf{t}_u,\mathbf{t}_d)$ to Problem 7 is also optimal for Problem 6, i.e., $T_F^*(\mathbf{t}_u,\mathbf{t}_d) = T_F(\mathbf{S}^*(\mathbf{t}_u,\mathbf{t}_d),\mathbf{t}_u,\mathbf{t}_d)$.

By Lemma 2, instead of solving Problem 6, we can focus on solving Problem 7. Johnson's algorithm [14] (of complexity $O(K \log K)$) can guarantee to find an optimal sequence for the three-stage flow-shop scheduling problem in the special case where

$$\min_{k \in \mathcal{K}} \{t_{u,k}\} \geq \max_{k \in \mathcal{K}} \{t_{e,k}\}. \tag{18}$$

Note that (18) usually holds, as the execution duration of each task at the BS is usually small due to the strong computing capability at the attached MEC servers. Thus, we can use Johnson's algorithm to solve Problem 7 approximately. If (18) holds, the obtained solution is optimal; otherwise, it is usually a sub-optimal solution with close-to-optimal performance.

Next, we focus on solving Problem 5. Note that $\widetilde{T}_F(\mathbf{S}, \mathbf{t}_u, \mathbf{t}_d)$ in (17) is convex, implying that $\widetilde{T}_F^*(\mathbf{S}, \mathbf{t}_u, \mathbf{t}_d)$ is convex. Thus, Problem 5 is convex and can be solved using standard convex optimization techniques. As $\widetilde{T}_F^*(\mathbf{S}, \mathbf{t}_u, \mathbf{t}_d)$ does not have a simple form, the complexity for solving Problem 5 may still be high. Let $T_{u,d}$ denote the total uploading and downloading duration. To reduce the complexity for solving Problem 5, we propose a related problem.

Problem 8 (Duration Allocation for Given $T_{u,d}$): For all $T_{u,d} > 0$, we have

$$\begin{aligned} \min_{\mathbf{t}_u, \mathbf{t}_d} \quad & E(\mathbf{t}_u, \mathbf{t}_d) \\ s.t. \quad & \sum_{k \in \mathcal{K}} (t_{u,k} + t_{d,k}) \leq T_{u,d}, \\ & (1), (3). \end{aligned}$$

Let $(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ denote the optimal solution.

Let $T_{u,d}^* \triangleq \sum_{k \in \mathcal{K}} (t_{u,k}^* + t_{d,k}^*)$, where $(\mathbf{t}_u^*, \mathbf{t}_d^*)$ is the optimal solution to Problem 5. As any uploading duration and downloading duration do not overlap, we can easily show that $(\mathbf{t}_u^\dagger(T_{u,d}^*), \mathbf{t}_d^\dagger(T_{u,d}^*)) = (\mathbf{t}_u^*, \mathbf{t}_d^*)$. That is, for optimal $T_{u,d}^*$ obtained by solving Problem 5, the optimal solution to Problem 8 with $T_{u,d}^*$ is the same as that to Problem 5. Later, the relationship between Problem 5 and Problem 6 will be used to reduce the complexity for solving Problem 5. We can easily verify that Problem 8 is convex and Slater's condition is satisfied, implying that strong duality holds. Thus, Problem 8 can be solved using KKT conditions. The optimal solution $(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ to Problem 8 is given by:

$$\begin{cases} t_{u,k}^\dagger(T_{u,d}) = \dfrac{L_{u,k} \ln 2}{B\left(W\left(\frac{\lambda^*(T_{u,d})|h_k|^2 - n_0}{n_0 e}\right) + 1\right)} \\ t_{d,k}^\dagger(T_{u,d}) = \dfrac{\beta L_{d,k} \ln 2}{B\left(W\left(\frac{\lambda^*(T_{u,d})|h_k|^2 - n_0}{n_0 e}\right) + 1\right)} \end{cases}, \ k \in \mathcal{K}, \tag{19}$$

where $W(\cdot)$ denotes the Lambert function, e is a mathematical constant that is the base of the natural logarithm, and $\lambda^*(T_{u,d})$ satisfies:

$$\sum_{k \in \mathcal{K}} \frac{(L_{u,k} + \beta L_{d,k}) \ln 2}{B\left(W\left(\frac{\lambda^*(T_{u,d})|h_k|^2 - n_0}{n_0 e}\right) + 1\right)} = T_{u,d}. \tag{20}$$

Note that, $\lambda^*(T_{u,d})$ in (20) can be easily obtained using the bisection method. Thus, we can compute $(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ efficiently. In addition, it can be easily verified that, $t_{u,k}^\dagger(T_{u,d})$ and $t_{d,k}^\dagger(T_{u,d})$ increase with $T_{u,d}$ for all $k \in \mathcal{K}$.

Based on the above analysis, we can obtain a sub-optimal solution to Problem 1. The details are summarized in Algorithm 1.

Algorithm 1 : Sub-optimal Solution to Problem 5

1: Set $T_{u,d} = T - \sum_{k \in \mathcal{K}} t_{e,k}$.
2: **repeat**
3: Solve Problem 8 by calculating $(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ according to (19) and (20).
4: Solve Problem 7 using Johnson's algorithm to obtain $\mathbf{S}^*(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ and then use (16) to obtain $T_F(\mathbf{S}^*(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d})), \mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$.
5: Set $T_{u,d} = \sum_{k \in \mathcal{K}} (t_{u,k}^\dagger(T_{u,d}) + t_{d,k}^\dagger(T_{u,d})) + (T - T_F(\mathbf{S}^*(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d})), \mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d})))$.
6: **until** $T_F(\mathbf{S}^*(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d})), \mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d})) = T$

Note that the time complexity for Step 3 (with $\lambda^*(T_{u,d})$ obtained using bisection search) is polynomial. The complexity for Step 4 is $O(K \log K)$. We can show that if (18) holds for $\mathbf{t}_u^\dagger(T_{u,d})$ where $T_{u,d} = T - \sum_{k \in \mathcal{K}} t_{e,k}$, Algorithm 1 converges.

By structural properties of Problem 7 and Problem 8, we can show the following result.

Lemma 3: If constraint (18) holds for $(\mathbf{t}_u^\dagger(T_{u,d}), \mathbf{t}_d^\dagger(T_{u,d}))$ given in (19), where $T_{u,d} = T - \sum_{k \in \mathcal{K}} t_{e,k}$, then $E(\mathbf{t}_u, \mathbf{t}_d)$ decreases as the number of iterations increases and Algorithm 1 stops in a finite number of iterations.

V. Simulation Results

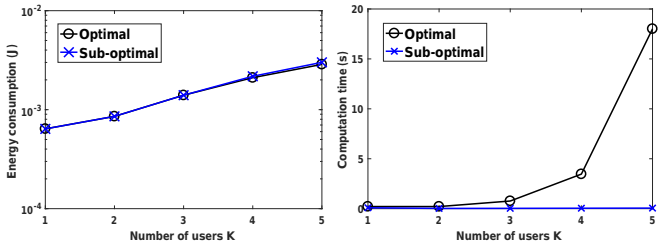
In the simulation, we consider the following settings [4], [5]. We let $\beta = 0.1$, $T = 80$ms, $\mu = 10^{-29}$, and $F = 6 \times 10^9$. Channel power gain h_k for mobile k is modeled as Rayleigh fading with average power loss 10^{-3}. The complex additive white Gaussian channel noise is $n_0 = 10^{-9}$ W. For each task k, $L_{u,k}$ and $L_{d,k}$ follow the uniform distribution over $[1 \times 10^5, 5 \times 10^5]$ (bits), and N_k follows the uniform distribution over $[0.5 \times 10^7, 1.5 \times 10^7]$ (CPU-cycles). All random variables are independent.

A. Comparison Between Optimal and Sub-optimal Solutions

In this part, we use a numerical example to compare the optimal solution and the proposed sub-optimal solution in terms of the weighted sum energy consumption and computational complexity. From Fig. 4(a), we can see that the performance of the proposed sub-optimal solution is very close to that of the optimal solution. From Fig. 4(b), we can see that the computation time for computing the sub-optimal solution grows at a much smaller rate than the optimal solution with respect to the number of users. This numerical example demonstrates the applicability and efficiency of the sub-optimal solution.

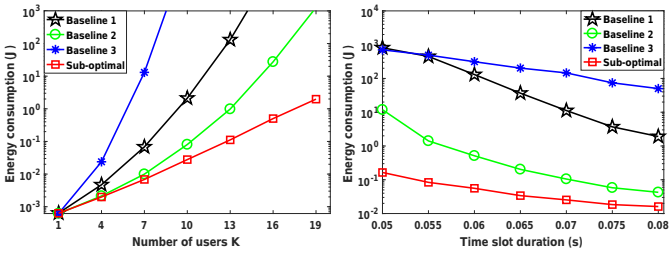
B. Comparisons with Existing Schemes

In this part, we compare the proposed sub-optimal solution (using Algorithm 1) with three existing comparison schemes.



(a) K vs energy consumption. (b) K vs computation time.

Fig. 4: Comparison between optimal and sub-optimal solutions.



(a) K users at $T = 80$ms. (b) Deadline T at $K = 10$.

Fig. 5: The weighted sum energy consumption versus the number of users K and the deadline T for the multi-user MEC system with non-negligible task execution durations.

All the three baseline schemes assume that the transmission and execution durations cannot be paralleled, and consider $T - \sum_{k=1}^{K} t_{e,k}$ as the total uploading and downloading duration. In particular, Baseline 1 allocates the total uploading and downloading duration $T - \sum_{k=1}^{K} t_{e,k}$ equally to the uploading and downloading operations of all tasks, i.e., $t_{u,k} = t_{u,k} = \frac{T - \sum_{k=1}^{K} t_{e,k}}{2K}$ for all $k \in \mathcal{K}$ [5]. Baseline 2 optimally allocates the total uploading and downloading duration to uploading and downloading operations to minimize the weighted sum energy consumption, using the optimal solution in (19) and (20). Baseline 3 allocates the total uploading and downloading duration to all users according to their task sizes (here we ignore the size of each computation result) to minimize the weighted sum energy consumption, using the optimal solution in (19) and (20), and then for each user, allocates the time duration to its uploading and downloading duartion operations to minimize the energy consumption corresponding to its task.

Fig. 5(a) and Fig. 5(b) illustrate the weighted sum energy consumption versus the number of users K and the deadline T, for the sub-optimal solution and the baseline schemes. From Fig. 5(a) and Fig. 5(b), we can observe that as the number of users K increases or the deadline T decreases, the weighted sum energy consumption increases. Baseline 3 achieves the maximum weighted energy consumption among all the three schemes, as it does not properly consider the size of the computation result of each task. The sub-optimal solution greatly outperforms the three baselines, as it more efficiently utilizes the time duration over which the transmission and execution operations are conducted in parallel (and hence maximizes the total transmission time).

VI. Conclusion

In this paper, we consider energy-efficient resource allocation schemes for a multi-user mobile edge computing system with inelastic computation tasks and non-negligible task execution durations. We first establish a computation-offloading model with non-negligible task execution durations. Then, we formulate the weighted sum energy consumption minimization problem by optimally allocating communication and computation resources, which is NP-hard in general. We propose a method to obtain an optimal solution and develop a low-complexity algorithm to obtain a sub-optimal solution, by connecting the optimization problem to a three-stage flow-shop scheduling problem and utilizing Johnson's algorithm as well as convex optimization techniques. Finally, numerical results show that the proposed sub-optimal solution outperforms existing comparison schemes significantly.

References

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

[2] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Select. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[3] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE ISIT*, July 2016, pp. 1451–1455.

[4] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *in Proc. IEEE WCNC*, Mar. 2017, pp. 1–6.

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

[6] Y. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and cpu time allocation for mobile edge computing," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–6.

[7] J. Cheng, Y. Shi, B. Bai, and W. Chen, "Computation offloading in cloud-ran based mobile cloud computing system," in *Proc. IEEE ICC*, May 2016, pp. 1–6.

[8] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[9] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," in *Proc. IEEE ICC*, May 2016, pp. 1–6.

[10] Y. Cui, W. He, C. Ni, C. Guo, and Z. Liu, "Energy-efficient resource allocation for cache-assisted mobile edge computing," in *Proc. IEEE LCN*, Oct. 2017.

[11] H. Emmons and G. Vairaktarakis, *Flow shop scheduling: theoretical results, algorithms, and applications*. Springer Science & Business Media, 2012.

[12] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-user mobile edge computing," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1611.01786

[13] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing." *HotCloud'10*, vol. 10, pp. 4–4, 2010.

[14] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," *Naval research logistics quarterly*, vol. 1, no. 1, pp. 61–68, 1954.

[15] H. D. Mathes, "A 2-machine sequencing problem with machine repetition and overlapping processing times," *OR-Spektrum*, vol. 21, no. 4, pp. 477–492, 1999.

[16] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power cmos digital design," *IEICE Trans. Electronics*, vol. 75, no. 4, pp. 371–382, 1992.