IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 1, 2019, accepted September 11, 2019. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.2944184

Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing

RYUJI KOBAYASHI AND KOICHI ADACHI, (Member, IEEE)
Advanced Wireless and Communication Research Center, The University of Electro-Communications, Tokyo 182-8585, Japan
Corresponding author: Koichi Adachi (adachi@awcc.uec.ac.jp)

This work was supported by the European Commission in the framework of the H2020-EUJ-02-2018 project 5G-Enhance (Grant agreement no. 815056) and the Ministry of Internal Affairs and Communications (MIC) of Japan.

ABSTRACT Due to explosive growth in mobile applications and services with different requirements, the concept of mobile edge computing (MEC) has emerged. For MEC, a mobile user (MU) and a MEC server need to exchange tasks using limited radio resources. Furthermore, when multiple MUs possess tasks, the MEC server has to handle multiple tasks simultaneously. Thus, the radio and computing resources need to be allocated to MUs by taking into account the wireless channel condition and the computing power of MUs and the MEC server. In this paper, a radio resource and computing resource allocation scheme is proposed to minimize the total processing completion time of all the tasks. Each task is assumed to be divided into *local task* and *offload task*. The local task is processed by each MU while the offload task is processed by a MEC server. We first formulate the optimization problem to minimize the total processing completion time of all tasks. To solve the formulated optimization problem, we propose a two-step radio and computing resources allocation scheme which iteratively performs bisection search method and Johnson's algorithm. The numerical results elucidate that the proposed scheme can reduce the total processing completion time by about 25% on average compared to the conventional schemes when multiple MUs have divisible tasks.

INDEX TERMS Mobile edge computing, resource allocation, optimization.

I. INTRODUCTION

Mobile devices, such as smartphones, are inevitable in daily life. Along with the development of such mobile devices, many new mobile applications such as the Internet, video stream analysis, augmented reality (AR) and object authentication (e.g., face authentication) are emerging. However, since mobile devices have resource constraints (limited computing power and storage capacity), it is challenging to compute heavy application tasks that require high computing power. Thus, such applications are difficult to be processed on mobile devices only. To tackle this problem, the concept of mobile cloud computing (MCC) has been proposed [1]. In the MCC, a mobile device can utilize computing resources and storage resources of a powerful remote centralized cloud (CC) by offloading its task. The remote CC is accessible by mobile devices via the mobile operator's core network and the Internet [1], [2]. Thus, MCC improves the user experience by extending the capability of mobile devices [3]. However, since the task to be processed by remote CC needs to be exchanged between the cloud server and the mobile device, MCC incurs significant delay [4]. Furthermore, as the number of tasks to be processed increases, they consume the transmission capacity of the backbone network.

Due to the reasons as mentioned above, mobile edge computing (MEC) technology that moves cloud computing power and storage capacity to the edge of the radio access network (RAN) such as base stations (BSs) has attracted attention [2]. Due to the proximity to mobile devices, MEC can significantly reduce processing time compared to MCC. However, the *offload tasks* still need to be exchanged over the wireless link between a BS equipped with MEC server and mobile users (MUs). Since the radio resources in a wireless link such as frequency and time are limited, it is necessary to utilize them efficiently. When multiple MUs try to offload their tasks to the same MEC server, the radio resource and computing

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

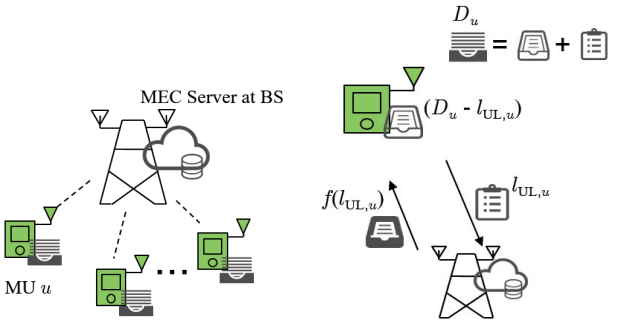
resources of the MEC server must be shared by multiple MUs. Thus to fully enjoy the benefit of MEC, the efficient management of radio resource and computing resource of MEC server is important.

The finite battery lifetime of mobile devices, limited communication and computing resources become a problem in designing MEC systems with high energy efficiency due to latency requirements of applications. The main objective of many works done so far is the design of task offloading strategy and resource management scheme such as allocation of radio and computing resources to each MUs [5]. The problem of offloading strategy and resource allocation related to MEC has been considered in the vehicle network [6], [7], the virtualization network [8], and the cellular network [9]–[22]. In [6], [8]–[13], the offloading strategy is tackled. In [7], [14]–[22], the resource allocation problem is tackled. In the past, the allocations of radio and computing resources were separately considered. However, separately considering the allocations of radio and computing resources may result in resource congestion, which leads to a significant waste of the other resource [15]. Therefore, the allocations of radio and computing resources need to be considered jointly. An optimization problem for minimizing the completion time of *each* task has been formulated [16]. A resource allocation scheme for maximizing the energy efficiency of MEC has been proposed in [17]. The conventional works mainly aim at minimizing completion time or minimizing the energy consumption of each task or MU.

Furthermore, in most of the existing works, the size of the result after computing at the MEC server is assumed to be negligible and ignored. Thus, the impact of transmission of computing results on the downlink (DL), the transmission from BS equipped with MEC to MU, has not been taken into account. However, the size of the computing result may become too large to be ignored. For example, the computing result size after computing at the MEC server for the VR application is large, so the amount of communication data increases, leading to a delay [7]. In [7] and [18], resource allocation using certain application characteristics such as virtual reality (VR) and AR are considered. If the computing result size is large, DL time may not be negligible anymore. In that case, DL time has a significant impact on the total processing completion time. Since the local computation is also considered in this manuscript, the additional time introduced by DL transmission for the offload task provides additional computing time for the local task. Thus, the optimal task scheduling may be different from the one ignoring DL transmission. Besides, the local computing at MU is not considered in many previous works. Thus, it is assumed that the task of each MU is to be offloaded entirely. Completely offloading all tasks leads to exhaustion of radio and computing resources. Therefore, it is important to consider both computing capabilities of the MU and the MEC server. By computing each task partially at both MU and MEC, the more efficient utilization of radio and computing resources can be achieved. In a very recent work [22], a non-orthogonal multiple access (NOMA) based partial task offloading strategy has been proposed. It has been shown that by taking advantage of NOMA, the total processing completion time can be shortened compared to time division multiple access (TDMA) based offloading. However, it is necessary to form a cluster of MUs that transmit and receive signals simultaneously, and successive interference cancellation (SIC) is mandatory at MUs.

In this paper, an efficient radio and computing resource allocation scheme is proposed to minimize *the total processing completion time of multiple tasks*. To consider the application to delay-sensitive tasks, in this paper, the system where there is a central controller that decides the task offloading is considered as in literature [9]–[22]. The proposed scheme has simple since it only decides the size of the offloading task of each MU. No additional complexity or signal processing is required at the MU. Here, the total processing completion time is defined as the time at which offload processing and local computing at all MUs are completed. Firstly, resource allocation is formulated as a mixed discrete-continuous optimization problem that is an NP-hard problem. Secondly, to solve it, we propose a two-step resource allocation algorithm that iteratively solves the sub-problems. In the first sub-problem, the time durations for uploading, processing at MEC, and downloading, are determined in order to minimize the processing time of *each MU*. Then, in the second sub-problem, the processing order among *multiple MUs* is determined by considering it as a *flow shop scheduling problem*. The performance comparison of the proposed scheme against the existing works is performed by computer simulation. The numerical evaluation elucidates that the proposed scheme can reduce the total processing completion time of all the tasks on average by about 25% compared to the existing works. Furthermore, the influence of the wireless duplex system on the total processing completion time of the proposed scheme is discussed. In the prior work [16], UL transmission is performed by either time division or frequency division, and processing by MEC server is performed frequency division. On the other hand, in this work, all the processing is performed by time division, and DL transmission is also taken into account. In [17], an MU itself does not perform any task processing. Thus, all the tasks are offloaded to the MEC server. However, as the number of MUs increases, the radio resources required for UL/DL may become the bottleneck for task processing completion time. Thus, in this paper, in consideration of local processing, each MU divides the computing task into offload task and local task and performs task processing, which leads to the effective use of radio resources and computation resources.

The remainder of this paper is organized as follows. Section 2 describes the system model. After describing the task model and the processing time required for transmission and computation, an optimization problem is formulated to minimize the total processing completion time of task in Section 3. In Section 4, a two-step algorithm using bisection search and Johnson's algorithm is proposed to solve the formulated optimization problem. Section 5 examines the



(a) Mutiuser MEC system (b) Task model

FIGURE 1. System model.

performance of the proposed scheme by computer simulation. Finally, Section 6 concludes this paper.

II. SYSTEM MODEL

A multiuser MEC system with MUs $\mathcal{U} = \{1, 2, \ldots, U\}$ and one BS with MEC server is considered as shown in Fig. 1. Each MU with its own local computing power has a task. We assume that each task can be divided into local task and offload task based on the resource allocation scheme [16]. The local task is to be processed by the local CPU at each MU. The offload task is to be transmitted to the MEC server in the UL and processed. The computing result is transmitted from the MEC server to each MU in the DL.

A. TASK MODEL AND PROCESSING TIME

The task size of MU $u \in \mathcal{U}$ is denoted by D_u [bits], and the task type is denoted by A_u [CPU cycles/bit]. The task type indicates the number of CPU cycles required to process one bit of task. Based on the result of resource allocation, each task is divided into the local task and the offload task. Letting the size of the offload task be $l_{\mathrm{UL},u}$ [bits], the size of the local task can be expressed as $(D_u - l_{\mathrm{UL},u})$ [bits]. In this paper, the size of the computing result of the offloaded task is modeled as $f(l_{\mathrm{UL},u})$ [bits] which is a function of offloaded task size $l_{\mathrm{UL},u}$.

The processing of the offload task is composed of three phases: uploading the task, computing the task at the MEC server, and returning the computation result from the MEC server to the MU. Different from the prior works such as [16], the completion time to finish all the tasks is minimized. Each MU is assumed to possess one distinct task, which can be divided into two parts, i.e., *local task* and offload task. The local task is processed by each MU locally while the offload task is processed by the MEC server. For the offload task, the task itself needs to be transferred from each MU to the MEC server via the uplink (UL) channel. Once the task is processed by the MEC server, the output needs to be returned from the MEC server to each MU via the DL channel. Thus, it is necessary for the MEC server to appropriately allocate the upload time, the computing time, and the download time for each MU. The proposed resource allocation scheme is composed of two steps. In the first step, the time durations for upload, task computing, and download are determined by Bisection search method. Once the time durations are determined, a flow shop scheduling problem is applied [17]. The upload, computing, and download of each task are scheduled by Johnson's algorithm in order to minimize the total processing completion time of all MUs.

1) LOCAL TASK COMPUTING TIME

Let the computing capability of MU u be $F_{\mathrm{local},u}$ [CPU cycles/sec]. Since MU u needs to compute the task of $(D_u - l_{\mathrm{UL},u})$ [bits] locally, computing time $t_{\mathrm{local},u}$ of the local task is given by

$$t_{\mathrm{local},u} = \frac{A_u(D_u - l_{\mathrm{UL},u})}{F_{\mathrm{local},u}}, \quad \forall u \in \mathcal{U}. \tag{1}$$

2) OFFLOAD TASK PROCESSING TIME

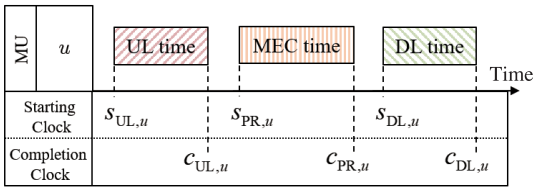
The offload task is first transmitted to the MEC server in the UL, and processed by the MEC server, and then the result is received in the DL. Thus, the processing time of the offload task consists of uploading time, computing time at MEC, and downloading time. The processing time, $t_{\mathrm{offload},u}$, taken for MU u is given by

$$t_{\mathrm{offload},u} = \frac{l_{\mathrm{UL},u}}{x_u C_{\mathrm{UL},u}} + \frac{A_u l_{\mathrm{UL},u}}{F_{\mathrm{mec}}} + \frac{f(l_{\mathrm{UL},u})}{y_u C_{\mathrm{DL},u}}, \quad \forall u \in \mathcal{U}. \tag{2}$$

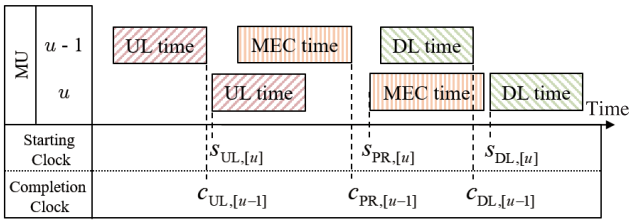
where $C_{\mathrm{UL},u}$ [bits/s] and $C_{\mathrm{DL},u}$ [bits/s] are UL and DL channel capacity of MU u. The ratio of UL and DL allocated to the MU be $0 \leq x_u \leq 1$ and $0 \leq y_u \leq 1$, respectively. In addition, F_{mec} [CPU cycles/s] is the task computing capability of MEC server. In (2), the first term is the time required to send the offload task to the MEC, the second term is the time required for computing offload task at the MEC server, and the third term is the time required to return the task processed by the MEC server to the MU.

B. SCHEDULING

In this paper, the scheduling of uploading, computing, and downloading tasks of MUs are executed in a time-division manner. Furthermore, it is assumed that the local processing starts at when uploading starts. To specify this model mathematically, we define the same notation and constraints as in [17]. Let $s_{\mathrm{UL},u}$, $s_{\mathrm{PR},u}$, and $s_{\mathrm{DL},u}$ denote the starting clocks for uploading, computing at MEC server, and downloading offload task of MU u, respectively. In addition, let $c_{\mathrm{UL},u}$, $c_{\mathrm{PR},u}$, and $c_{\mathrm{DL},u}$ denote the completion clocks for uploading, computing at MEC server, and downloading offload task of MU u, respectively. Fig. 2 shows the order constraints for the offload task for each MU and multiple MUs.



(a) Order constraint I for each MU



(b) Order constraint II among multiple MUs

FIGURE 2. Constraint on task processing order.

1) COMPLETION TIME CONSTRAINT OF OFFLOAD MODEL

The time constraint in each process of offload task u is expressed as follows:

$$\begin{cases} s_{\mathrm{UL},u} + \dfrac{l_{\mathrm{UL},u}}{x_u C_{\mathrm{UL},u}} = c_{\mathrm{UL},u} \\ s_{\mathrm{PR},u} + \dfrac{A_u l_{\mathrm{UL},u}}{F_{\mathrm{mec}}} = c_{\mathrm{PR},u} \\ s_{\mathrm{DL},u} + \dfrac{f(l_{\mathrm{UL},u})}{y_u C_{\mathrm{DL},u}} = c_{\mathrm{DL},u} \end{cases}, \forall u \in \mathcal{U}. \tag{3}$$

2) CONSTRAINT ON TASK PROCESSING ORDER I

As Fig. 2(a) shows there are constraint conditions for the offload task for each MU. The constraint conditions for the uploading, the computing at MEC server, and the downloading of the computing result are expressed as

$$\begin{cases} s_{\mathrm{UL},u} \geq 0 \\ s_{\mathrm{PR},u} \geq c_{\mathrm{UL},u} \\ s_{\mathrm{DL},u} \geq c_{\mathrm{PR},u} \end{cases}, \forall u \in \mathcal{U}. \tag{4}$$

Equation (4) indicates that each processing of offload task cannot be interrupted by other processing.

3) CONSTRAINT ON TASK PROCESSING ORDER I

As Fig. 2(b) shows there are constraint conditions among MUs. The uploading, the computing at MEC server, and the downloading of the computing result for each MU should be exclusive. Thus, we have

$$\begin{cases} s_{\mathrm{UL},[u]} \geq c_{\mathrm{UL},[u-1]} \\ s_{\mathrm{PR},[u]} \geq c_{\mathrm{PR},[u-1]} \\ s_{\mathrm{DL},[u]} \geq c_{\mathrm{DL},[u-1]} \end{cases}, \forall u \in \mathcal{U} \backslash \{1\}, \tag{5}$$

where subscript $[u]$ is the task index at position u of sequence $\mathbf{s}$. Equation (5) ensures that at most one task is executed at any time and at most one task is transmitted. Here, it is possible to guarantee not to receive in DL until UL transmission of all the tasks is completed, i.e., time division duplex (TDD), by adding the constraint represented by

$$s_{\mathrm{DL},[1]} \geq c_{\mathrm{UL},[U]}. \tag{6}$$

In the following, we consider TDD only. The impact of the duplexing on the performance of the proposed scheme will be evaluated in Section 5.

III. PROBLEM FORMULATION

The task processing completion time, $T_{\mathrm{comp},u}$, of MU $u \in \mathcal{U}$ is defined as the time when both the processing of the local task and the offload task are completed, which can be expressed as

$$T_{\mathrm{comp},u} \triangleq \max\{t_{\mathrm{local},u}, c_{\mathrm{DL},u}\}, \quad \forall u \in \mathcal{U}, \tag{7}$$

where $t_{\mathrm{local},u}$ and $c_{\mathrm{DL},u}$ are given by (1) and (3), respectively, and $\max\{\cdot,\cdot\}$ returns the maximum value of the arguments. Thus, the total processing completion time of all MUs is given by

$$T = \max_{1 \leq u \leq U} T_{\mathrm{comp},u}. \tag{8}$$

In this paper, we consider the problem of minimizing the total processing completion time of all MUs in multiuser MEC system. An optimization problem that minimizes T given by (8) can be formulated as

$$\min_{\substack{\{l_{\mathrm{UL},u}\},\{x_u\},\{y_u\},\\ \mathbf{s}\in\mathbb{S},\{s_{\mathrm{UL},u}\},\{s_{\mathrm{PR},u}\},\{s_{\mathrm{DL},u}\}}} \quad T \tag{P}$$

$$\text{subject to } t_{\mathrm{local},u} \leq T, \quad \forall u \in \mathcal{U}, \tag{9}$$

$$t_{\mathrm{offload},u} \leq T, \quad \forall u \in \mathcal{U}, \tag{10}$$

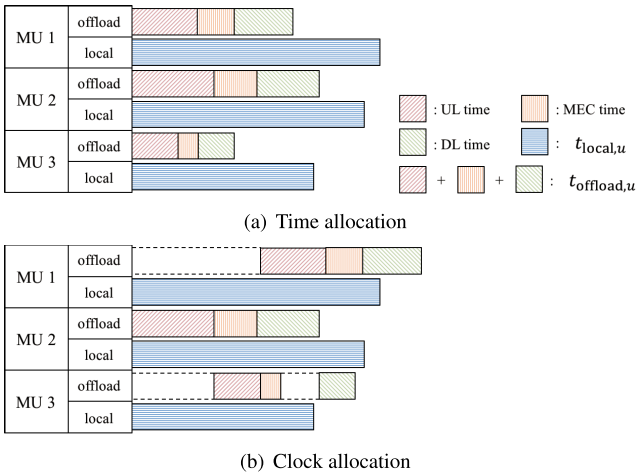
$$\sum_{u=1}^{U} x_u = 1, \quad x_u \geq 0, \ \forall u \in \mathcal{U}, \tag{11}$$

$$\sum_{u=1}^{U} y_u = 1, \quad y_u \geq 0, \ \forall u \in \mathcal{U}, \tag{12}$$

$$0 \leq l_{\mathrm{UL},u} \leq D_u, \quad \forall u \in \mathcal{U},$$

$$(3), (4), (5), (6) \tag{13}$$

where $\mathbb{S}$ indicates a set of feasible task schedules for all MUs. Equation (9) represents the constraint on the completion time of the local task, and (10) represents the constraint on the processing time of the offload task. Equations (11) and (12) are constraints on channel utilization of UL and DL, respectively. Equation (13) represents the constraint on the size of the offload task. Completion clocks $\{c_{\mathrm{UL},u}\}$, $\{c_{\mathrm{PR},u}\}$, and $\{c_{\mathrm{DL},u}\}$ are determined once their corresponding starting clocks $\{s_{\mathrm{UL},u}\}$, $\{s_{\mathrm{PR},u}\}$, $\{s_{\mathrm{DL},u}\}$, offload task size $\{l_{\mathrm{UL},u}\}$, and channel utilization $\{x_u\}$ and $\{y_u\}$ are decided. This optimization problem (P) allocates time by the time-division manner, assigns start and end clocks of processing in each phase, and performs scheduling. However, this optimization problem is NP-hard because of a mixed discrete-continuous optimization problem where the concept of time and clock



(a) Time allocation

(b) Clock allocation

FIGURE 3. Assign task time and clock of each MU.

is mixed [17]. Therefore, in this paper, we decompose the original optimization problem into two sub-problems and execute sub-optimal radio and computing resource allocation. In the first sub-problem, time allocation problem is dealt with. In the second sub-problem, clock allocation problem so called scheduling problem is dealt with. Fig. 3 shows how the time allocation and the clock allocation are performed in the first and the second sub-problem, respectively .

A. TIME ALLOCATION

As shown in Fig. 3(a), the processing time of task of MU u can be expressed by the following equation

$$T_{\text{comp},u}^{(\mathsf{P1})} = \max\{t_{\text{local},u}, t_{\text{offload},u}\}, \quad \forall u \in \mathcal{U}. \tag{14}$$

Thus, the worst processing time can be expressed as

$$T^{(\mathsf{P1})} = \max_{1\le u\le U} T_{\text{comp},u}^{(\mathsf{P1})}. \tag{15}$$

By assigning the task size and the time so that the worst processing time becomes shortest, it is possible to obtain the optimum resource for problem (P). Thus, we can obtain the time related optimization problem (P1), and by solving this problem we get the optimal resource with respect to time. The optimization problem (P1) is given by

$$\min_{\{l_{\text{UL},u}\},\{x_u\},\{y_u\}} \quad T^{(\mathsf{P1})}$$
$$\text{subject to} \ \ (9), (10), (11), (12), (13). \tag{P1}$$

B. CLOCK ALLOCATION (SCHEDULING)

In this paper, since the time-division manner is used, the uploading, the computing by MEC server, and downloading of each task cannot be performed at the same time. Furthermore, in TDD, since the same frequency band is assigned to UL and DL, there are restrictions on the timing of UL and DL. Thus, even if the computation of a task is completed at MEC, it cannot be returned by DL until all MUs complete UL transmission. Therefore, it is necessary to determine the clock allocation of problem (P). Once time allocation and clock scheduling are performed, as shown in Fig. 3(b), the completion clock of the last task can be obtained. The optimum value of problem (P) is obtained by allocating and rearranging the starting clock so that the completion clock of the last task is minimized. Thus, the clock related optimization problem is described as

$$\min_{\mathbf{s}\in\mathbb{S},\{s_{\text{UL},u}\},\{s_{\text{PR},u}\},\{s_{\text{DL},u}\}} c_{\text{DL},[U]}$$
$$\text{subject to} \ \ (3), (4), (5), (6). \tag{P2}$$

That is, problem (P1) is a time allocation problem, and the problem concerning the clock allocation is solved in problem (P2). The objective of original problem (P) is to minimize the total completion time among all tasks. In another word, it is to minimize the time at which the last task is downloaded to an MU. The total completion time of each task is affected by the time durations of uplink, processing at MEC, and downlink and the scheduling strategy. Thus, the objective of sub-problem (P2) matches with the min-max objective in (P). Solving these two sub-problems leads to the sub-optimal solution.

IV. PROPOSED SCHEME

A. BISECTION SEARCH METHOD

In order to solve problem (P1), the optimum value search is performed by bisection search on $T^{(\mathsf{P1})}$. For this, it is necessary to take into account the constraints (9)-(13), which results in a feasibility problem [23]. To solve the feasibility problem, the bisection search can be adopted [16]. Different from [16], the time taken for DL is also considered in this paper in addition to UL time and the computing time.

From (9), the following equation holds:

$$D_u - \frac{T^{(\mathsf{P1})}F_{\text{local}}}{A_u} \le l_{\text{UL},u}. \tag{16}$$

In order for local task with $(D_u - l_{\text{UL},u})$ [bits] to be completed within processing time $T^{(\mathsf{P1})}$, $l_{\text{UL},u}$ must be greater than or equal to $\left(D_u - \frac{T^{(\mathsf{P1})}F_{\text{local}}}{A_u}\right)$. Therefore, the minimum offloading task size, $l_{\text{UL},u}^{\min}$, can be expressed as

$$l_{\text{UL},u}^{\min} = \max\left\{0, D_u - \frac{T^{(\mathsf{P1})}F_{\text{local}}}{A_u}\right\}. \tag{17}$$

In order to ensure the feasibility of (17), it is necessary to solve the following feasibility problem:

$$\text{find}\{x_u\}, \{y_u\} \tag{P1A}$$
$$\text{subject to} \ \frac{l_{\text{UL},u}^{\min}}{x_u C_{\text{UL},u}} + \frac{A_u l_{\text{UL},u}^{\min}}{F_{\text{mec}}} + \frac{f(l_{\text{UL},u}^{\min})}{y_u C_{\text{DL},u}} \le T^{(\mathsf{P1})}, \quad \forall u \in \mathcal{U}, \tag{18}$$
$$\frac{A_u l_{\text{UL},u}^{\min}}{F_{\text{mec}}} \le T^{(\mathsf{P1})}, \quad \forall u \in \mathcal{U}, \tag{19}$$
$$\sum_{u=1}^{U} x_u = 1, \quad x_u \ge 0, \ \forall u \in \mathcal{U}, \tag{20}$$
$$\sum_{u=1}^{U} y_u = 1, \quad y_u \ge 0, \ \forall u \in \mathcal{U}. \tag{21}$$

Algorithm 1 Bisection Search Algorithm

1: Initialize: $T_{\text{low}} = 0$, $T_{\text{high}} = \max_{1\le u\le U} \frac{A_u D_u}{F_{\text{local}}}$, set ε
2: **while** $T_{\text{high}} - T_{\text{low}} \ge \varepsilon$ **do**
3: Set $T = \frac{T_{\text{high}}+T_{\text{low}}}{2}$, and calculate $l^{\min}_{\text{UL},u}$, a_u, b_u, c_u
4: **if** Feasibility conditions (22), (23), and (24) are satisfied **then**
5: Set $T_{\text{high}} = T$, and go to step2
6: **else**
7: Set $T_{\text{low}} = T$, and go to step2
8: **end if**
9: **end while**

This is a feasibility problem of finding $\{x_u\}$ and $\{y_u\}$ so that the objective is set to 0 and only need to validate the feasible set.

Theorem 1: Letting $a_u = \frac{l^{\min}_{\text{UL},u}}{C_{\text{UL},u}}$, $b_u = \frac{A_u l^{\min}_{\text{UL},u}}{F_{\text{mec}}}$, and $c_u = \frac{f(l^{\min}_{\text{UL},u})}{C_{\text{DL},u}}$, the necessary and sufficient conditions for feasibility problem (P1A) can be expressed as (see Appendix A)

$$b_u \le T^{(\text{P1})}, \quad \forall u \in \mathcal{U}, \tag{22}$$

$$\left(\sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{(T^{(\text{P1})} - b_u)}\right)^2 \le \left(1 - \sum_{u=1}^{U} \frac{a_u}{(T^{(\text{P1})} - b_u)}\right) \times \left(1 - \sum_{u=1}^{U} \frac{c_u}{(T^{(\text{P1})} - b_u)}\right), \tag{23}$$

$$\sum_{u=1}^{U} \frac{a_u}{(T^{(\text{P1})} - b_u)} \le 1, \ \sum_{u=1}^{U} \frac{c_u}{(T^{(\text{P1})} - b_u)} \le 1. \tag{24}$$

Proof: See Appendix A.

In the conventional work, the feasibility condition of (22) is not considered because the processing at the MEC is not performed by a time-division manner. However, in this paper, since processing at the MEC is also performed by a time-division manner, the feasibility conditions only for processing time in UL and DL are not sufficient. Therefore, the feasibility condition about the processing time at the MEC server is required that the processing at the MEC server does not exceed the processing time. This feasibility condition is denoted by (22). With this feasibility condition, the algorithm can always converge. The algorithm for solving (P1) is given by Algorithm 1.

B. JOHNSON'S ALGORITHM

For time allocated by Bisection search method, the clock is allocated by solving (P2). Problem (P2) can be regarded as a job scheduling problem that is how to process the tasks in multiple machines in what order to minimize the overall total processing completion time. Job scheduling can be classified into three types of flow shop problem, job shop problem, and open shop problem depending on the order of task processing [24]. Since, uploading, computing by MEC server, and returning computing result is regarded as three operations, (P2) can be regarded as a 3-stage flow shop scheduling problem [17]. The 3-stage flow shop scheduling problem handles all tasks in three machines in the same processing order (uploading, computing at MEC server, and downloading). Here, if $T^{(\text{P2})}$ is the minimum total processing completion time, $T^{(\text{P2})}$ can be obtained by optimizing $s_{\text{UL},u}$, $s_{\text{PR},u}$, $s_{\text{DL},u}$ under the constraints of (3), (4), (5) [17]. Therefore, letting $T^{(\text{P2A})}$ be the optimal solution of (P2), problem (P2) will be rewritten by following equation as a 3-stage flow shop scheduling problem [17]:

$$T^{(\text{P2A})} = \min_{\mathbf{s}\in\mathbb{S}} \quad T^{(\text{P2})}, \tag{P2A}$$

where $T^{(\text{P2})}$ is given by

$$T^{(\text{P2})} = \max\left\{ \max_{1\le i\le j\le U} \left\{ \left(\sum_{u=1}^{j} \frac{A_{[u]} l_{\text{UL},[u]}}{F_{\text{mec}}} - \sum_{u=1}^{j-1} \frac{f(l_{\text{UL},[u]})}{y_{[u]} C_{\text{DL},[u]}} \right) + \left(\sum_{u=1}^{i} \frac{l_{\text{UL},[u]}}{x_{[u]} C_{\text{UL},[u]}} - \sum_{u=1}^{i-1} \frac{A_{[u]} l_{\text{UL},[u]}}{F_{\text{mec}}} \right) \right\}, \sum_{u=1}^{U} \frac{l_{\text{UL},[u]}}{x_{[u]} C_{\text{UL},[u]}} \right\} + \sum_{u=1}^{U} \frac{f(l_{\text{UL},[u]})}{y_{[u]} C_{\text{DL},[u]}}. \tag{25}$$

The rationale of optimization problem (P2A) is as follows [25]. It is necessary to find the scheduling strategy to minimize the time gap between the ending time of a specific job at the first (second) machine and its starting time at the second (third) machine. This results in the optimum utilization of the three machines.

Since TDD is adopted as a duplexing technology, (25) has additional term that takes into account constraint (6). By relaxing constraint (6), we can treat (P2A) as a standard 3-stage flow shop scheduling problem. The 3-stage flow shop scheduling problem is generally NP-hard [17], [25]. (P2A) is a 3-stage flow shop scheduling problem, so that the optimal solution can be obtained by Johnson's algorithm [25] when a certain condition is satisfied. The optimal solution for (P2A) is also optimal for (P2), so find the optimal solution of (P2A) by Johnson's algorithm [17]. The 3-stage flow shop scheduling using three separate machines (uploading, computing at MEC server, and downloading) can obtain an optimal solution by using Johnson's algorithm if any of the following condition is satisfied:

$$\max_{u\in\mathcal{U}} \left(\frac{A_u l_{\text{UL},u}}{F_{\text{mec}}} \right) \le \min_{u\in\mathcal{U}} \left(\frac{l_{\text{UL},u}}{x_u C_{\text{UL},u}} \right), \tag{26}$$

$$\max_{u\in\mathcal{U}} \left(\frac{A_u l_{\text{UL},u}}{F_{\text{mec}}} \right) \le \min_{u\in\mathcal{U}} \left(\frac{f(l_{\text{UL},u})}{y_u C_{\text{DL},u}} \right). \tag{27}$$

When the above conditions are satisfied, the solution obtained is the optimal solution. Since the MEC server has powerful computing power, the computing time of each task usually becomes small, and the above conditions, (26) and (27), are usually satisfied, so the optimum solution is obtained [17]. However, if the above conditions are not satisfied, it becomes

a sub-optimal solution that is close to the optimum performance [17]. In Johnson's algorithm, a list of time required for task processing in each machine is first prepared. In the case of 3 machines, assume a virtual machine called 1st machine + 2nd machine, 2nd machine + 3rd machine, apply Johnson's algorithm in 2 machines. Find the processing with the minimum processing time from the created unordered task list. If the processing of the found minimum processing time is the time taken for preprocessing (1st machine + 2nd machine) in the virtual machine, the task is processed first. On the other hand, if it is the time taken for the post-processing (2nd machine + 3rd machine) in the virtual machine, the task is finally processed. When the task processing order is determined, the task is deleted from the task list. This operation is repeated until the order of all the tasks is determined. The simple example of how Johnson's algorithm works is provided in Appendix B.

C. PROPOSED SCHEME AND SUB-OPTIMAL SOLUTION

In this paper, the original optimization problem (P) is divided into sub-problems (P1) and (P2). Time allocation is performed in (P1) and clock allocation is performed in (P2), respectively, and their optimal solutions are obtained respectively. For that purpose, the bisection search method is used for time allocation, and Johnson's algorithm is used for clock allocation. Specifically, bisection search determines the time durations of local computing and offload processing (uploading, computing at MEC server, and downloading) for each MU so that the total processing time of each MU is minimized. Applying the flow shop scheduling problem to the acquired time durations of offload processing (uploading, computing at MEC server, and downloading), Johnson's algorithm schedules the sequence of offload processing of MUs.

By using these two algorithms, the allocation of radio and computing resources for minimizing the total processing completion time is performed by two steps of iterative processing. First of all, local computing time, upload time, computing time at the MEC, and return time are determined using the bisection search method. For the time durations of offload processing obtained by the bisection search method, the order of each task is determined using Johnson's algorithm. By performing the scheduling using Johnson's algorithm, the order of each task and the minimum total processing completion time at the time obtained are obtained. Then, the obtained total processing completion time is evaluated according to the feasibility condition. In addition, replace the minimum total processing completion time with feasibility condition (23) related to the processing time of the offload task and the minimum total processing completion time is updated. This process is repeated until the difference between the computing time of the local task and the processing time of the offload task becomes infinitely small. By doing so, the overall total processing completion time is minimized. Thus, the scheme for achieving the overall minimum total processing completion time proposed in this research is given by Algorithm 2.

Algorithm 2 Proposed Scheme

Initialize: $T_{\text{low}} = 0$, $T_{\text{high}} = \max_{1 \le u \le U} \frac{A_u D_u}{F_{\text{local}}}$, set ε
while $T_{\text{high}} - T_{\text{low}} \ge \varepsilon$ **do**
  Set $T = \frac{T_{\text{high}} + T_{\text{low}}}{2}$, and calculate $l^{\min}_{\text{UL},u}$, a_u, b_u, c_u
  if Feasibility condition (22) is satisfied **then**
    Execution of Johnson's algorithm
    if $c_{\text{DL},[U]} < T$ and feasibility condition (24) are satisfied **then**
      Set $T_{\text{high}} = T$, and go to step2
    else
      Set $T_{\text{low}} = T$, and go to step2
    end if
  else
    Set $T_{\text{low}} = T$, and go to step2
  end if
end while

TABLE 1. Simulation parameters for wireless communication.

Number of MUs in each cell, U	10, 20, 30, 40, 50, 100
Coverage area, $R_{\text{C}} \times R_{\text{C}}$	1×1[km^2]
Bandwidth, B	10, 20, 50, 100 [MHz]
Transmit power of MU, P_{MU}	23 [dBm]
Transmit power of BS, P_{BS}	43 [dBm]
UL maximum data rate, $R^{\text{UL}}_{\max}$	6.0 [bps/Hz]
DL maximum data rate, $R^{\text{DL}}_{\max}$	8.0 [bps/Hz]
Maximum data rate, $R_{\max}$	6.0 [bps/Hz]
PSD of AWGN, N_0	-174 [dBm/Hz]
Shadowing deviation, σ	6.0 [dB]
Number of trials	100,000

TABLE 2. Simulation parameters for MEC system.

Local CPU, F_{local}	$\{1, 2, \ldots, 10\} \times 10^8$ [CPU cycles/s]
MEC CPU, F_{mec}	$\{5, 10, 15, 20, 25, 30\} \times 10^9$ [CPU cycles/s]
Data size, D_u	$\{100, 150, \ldots, 300\}$ [kbits]
Task type, A_u	$\{5, 6, \ldots, 15\} \times 10^2$ [CPU cycles/bit]
ε	0.0001
Computing result size, $f(l_{\text{UL},u})$	$\{l_{\text{UL},u}, 0.5 \times l_{\text{UL},u}, 2 \times l_{\text{UL},u}\}$

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed scheme. The simulation parameters for wireless communication are given in Table 1 and those for MEC system are given in Table 2. The system consists of 9 rectangular cells with a size of $R_{\text{C}} \times R_{\text{C}} = 1 \times 1$ [km^2] as shown in Fig. 4. A BS equipped with MEC server is located at the center of each cell. U MUs are randomly and uniformly distributed within each cell. The transmit power of each MU and BS are set to $P_{\text{MU}} = 23$ [dBm] and $P_{\text{BS}} = 43$ [dBm], respectively. The total frequency bandwidth is B [MHz]. TDD is adopted as a duplex technique except for Figs. 11 and 12, in which the impacts of MEC CPU F_{mec} and fraction of the bandwidth allocated to UL η on the performance in FDD system will be evaluated.

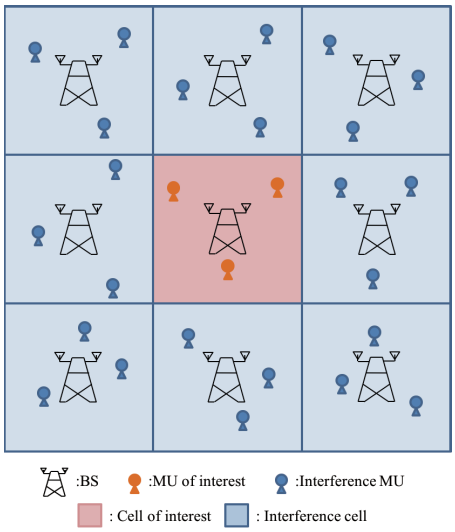


FIGURE 4. Interference model.

We consider the distance dependent pathloss and the log-normally distributed shadowing loss with a standard deviation σ [dB]. Let d [km] be the distance between a transmitter and a receiver, the propagation loss $L(d)$ [dB] is given by [26]

$$L(d) = 128.1 + 37.6 \log_{10}(d). \tag{28}$$

Considering the interference from interference cells, the channel capacity of MU u of the UL and DL in the cell of interest, which is the center cell, can be obtained from the Shannon channel capacity formula as

$$C_{\mathrm{UL},u} = B \times \max\{\log_2(1 + \mathrm{SINR}_u^{\mathrm{UL}}),\ R_{\max}^{\mathrm{UL}}\}, \tag{29}$$
$$C_{\mathrm{DL},u} = B \times \max\{\log_2(1 + \mathrm{SINR}_u^{\mathrm{DL}}),\ R_{\max}^{\mathrm{DL}}\}. \tag{30}$$

Here, B denotes the total channel bandwidth of system, $R_{\max}$ [bps/Hz] is the maximum spectrum efficiency of the system which is determined by the modulation and coding scheme (MCS). $\mathrm{SINR}_u^{\mathrm{UL}}$ and $\mathrm{SINR}_u^{\mathrm{DL}}$ are signal-to-interference plus noise power ratio (SINR) of UL and DL, respectively, and are expressed as

$$\mathrm{SINR}_u^{\mathrm{UL}} = \frac{P_{\mathrm{MU}} h_u}{P_{\mathrm{MU}} \frac{\sum_{u' \in \mathcal{U}'} h_{u'}}{U} + BN_0}, \tag{31}$$
$$\mathrm{SINR}_u^{\mathrm{DL}} = \frac{P_{\mathrm{BS}} h_u}{P_{\mathrm{BS}} \sum_{k \in \mathcal{K}} h_{k,u} + BN_0}, \tag{32}$$

where h_u denotes the channel gain of the BS in the cell of interest to MU u. Let $\mathcal{U}' = \{1, 2, \ldots, U'\}$ be set of the MUs in the interference cell and $h_{u'}$ denotes the channel gain of interference MU u' to the BS in the cell of interest. In addition, let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of BSs of the interference cell and $h_{k,u}$ denotes the channel gain of interference BS k to the MU u in the cell of interest. N_0 denotes the power spectral density of the additive white Gaussian noise (AWGN). Here, the total channel gain from the interference MU is divided by the number of MU U in the cell of interest. The reason for dividing by U is that the total channel gain from the interference MU is averaged by the number of MUs in the cell of interest.

The CPU frequency of local F_{local} and MEC server F_{mec}, data size D_u, and task type A_u are set with reference to [16]. The local CPU frequency $F_{\mathrm{local},u}$ is randomly selected from the set $\{1, 2, \ldots, 10\} \times 10^8$ [CPU cycles/s]. For each MU, the data size D_u and task type A_u are randomly selected from the sets $\{100, 125, 150, \ldots, 300\}$ [kbits] and $\{5, 6, \ldots, 15\} \times 10^2$ [CPU cycles/bit], respectively. For bisection search, ϵ is set to 10^{-4}.

TABLE 3. Time complexity of each algorithm for time allocation.

Algorithm	Time complexity
Exhaustive search	$O\left(\left(T_{\mathrm{high}} \times \frac{1}{\varepsilon}\right)\right)$
Bisection search method	$O\left(\log\left(T_{\mathrm{high}} \times \frac{1}{\varepsilon}\right)\right)$

TABLE 4. Time complexity of each algorithm for clock allocation.

Algorithm	Time complexity
Exhaustive search	$O(M!)$
Johnson's algorithm	$O(M \log M)$

A. EVALUATION ON COMPLEXITY OF PROPOSED SCHEME AND ITS CONVERGENCE PROPERTY

The proposed radio and computing resource allocation scheme iteratively performs the bisection search and Johnson's algorithm. Thus, an additional calculation is introduced. The lager the complexity is, the longer it takes to solve the problem. As a result, even if the optimum resource allocation can be obtained by the proposed scheme, the additional complexity may negatively offset the total processing completion time. Thus, first of all, we evaluate the impact of the additional complexity, which results from the executing algorithm itself, on the total processing completion time. It is assumed that the CPU F_{mec} of the MEC server is used as the value of the CPU required for calculating the time complexity when executing the algorithm of the proposed scheme.

1) TIME COMPLEXITY ON TIME ALLOCATION

Table 3 shows the time complexity of each algorithm for time allocation. The time complexity of the bisection search method is represented by $\mathcal{O}(\log_2 n)$ where n is the number of elements. Since the bisection search method is repeatedly executed until a certain criterion is met, the time complexity is expressed as $\mathcal{O}\left(\log_2\left(T_{\mathrm{high}} \times \frac{1}{\varepsilon}\right)\right)$ using the initial value, $T_{\mathrm{high}} = \frac{A_u D_u}{F_{\mathrm{local}}}$, and ε as the number of elements.

2) TIME COMPLEXITY ON CLOCK ALLOCATION

Table 4 shows the time complexity of each algorithm for clock allocation. The time complexity of Johnson's algorithm depends on the number of jobs, and the number of virtual machines does not affect. This is because the order of the virtual machines is fixed, and only the job order is manipulated.

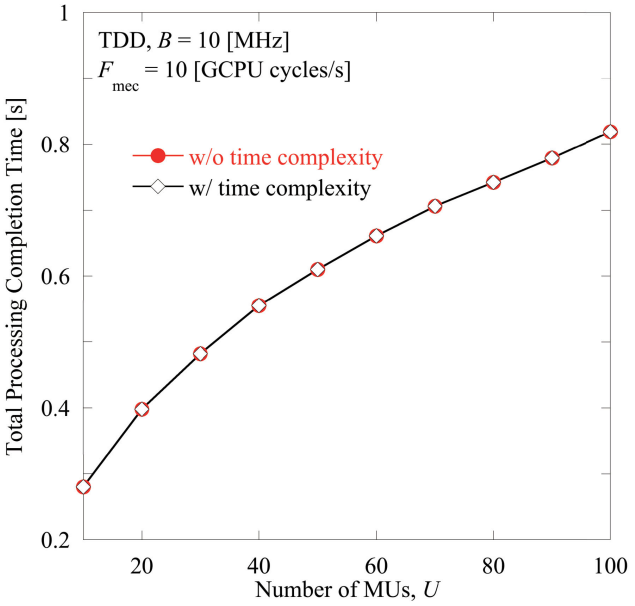


FIGURE 5. Comparison of average total processing completion time in consideration of time complexity.

The time complexity of Johnson's algorithm is expressed as $\mathcal{O}\left(M \log_2 M\right)$ where M is the number of MUs who offload tasks.

Fig. 5 shows the average total processing completion time of the proposed scheme as a function of number of MUs U. As shown in Table 3 and 4, the time complexity required to execute the algorithm increases as U increases. However, as Fig. 5 shows the total processing completion time of the proposed scheme is not affected by the time complexity required for execution. Thus, the time complexity required for executing is omitted in the following evaluation.

Fig. 6 shows the convergence property of the proposed scheme and the results of feasibility check, given by (22) and (24), for a specific realization of the channel conditions and tasks when $U = 10$, $B = 10$ [MHz], and $F_{\text{mec}} = 10 \times 10^9$ [CPU cycles/s] $= 10$ [GCPU cycles/s]. As can be seen from the figure that the proposed scheme converges to the sub-optimal value after a small number of iterations.

B. EVALUATION OF PROPOSED SCHEME

The total processing completion time of the proposed scheme is evaluated against the following resource allocation schemes:

1) All local: all tasks are processed locally.
2) All MEC: all tasks are offloaded and processed by MEC server.
3) All MEC w/ Johnson: all tasks are processed by MEC server only and task processing is scheduled using Johnson's algorithm.
4) Only Bisection: the radio and computing resources are allocated by bisection search method. Different from [16], computing at MEC server is performed in time division manner and DL is taken into account.

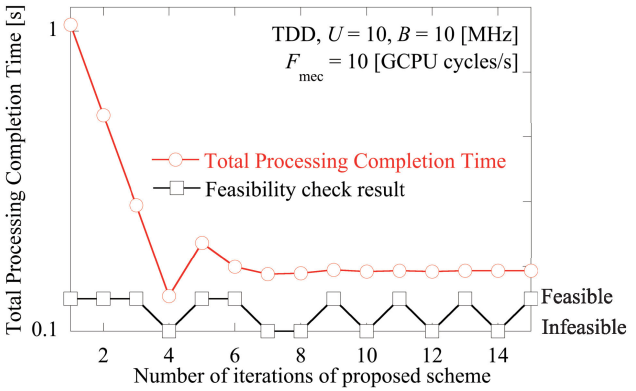


FIGURE 6. Convergence property of the proposed scheme.

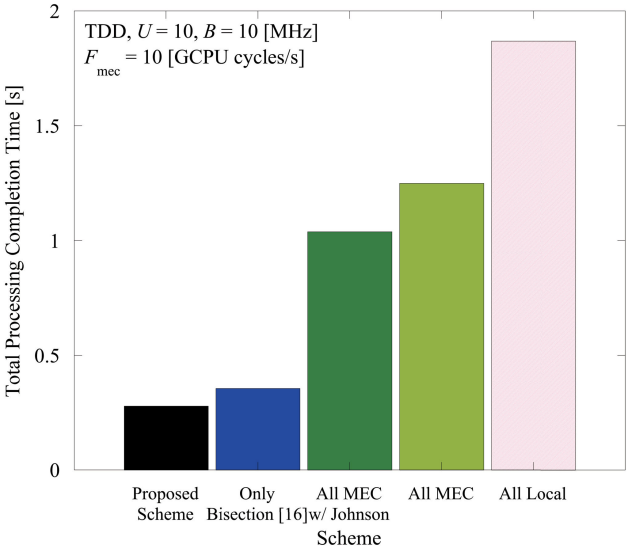


FIGURE 7. Comparison of average total processing completion time.

Fig. 7 shows the average total processing completion time of each scheme with $U = 10$, $B = 10$ [MHz], and $F_{\text{mec}} = 10$ [GCPU cycles/s]. As shown in Fig. 7, offloading tasks to MEC server using the bisection search method or Johnson's algorithm or both reduce the total processing completion time as compared to when all the tasks are processed locally. In addition, compared to the conventional scheme [16], the proposed scheme achieves the reduction of the total processing completion time by about 22% on average. This is because the use of Johnson's algorithm makes it possible to allocate resources considering clock allocation.

Fig. 8 shows the average total processing completion time in each scheme as a function of U with $B = 10$ [MHz] and $F_{\text{MEC}} = 10$ [GCPU cycles/s]. As U increases, the difference in the total processing completion time of schemes expands. When the number of MUs is $U = 100$, the proposed scheme reduces total processing completion time by about 25% compared to the conventional scheme.

Fig. 9 shows the average total processing completion time in each scheme in the case of the number of MUs is

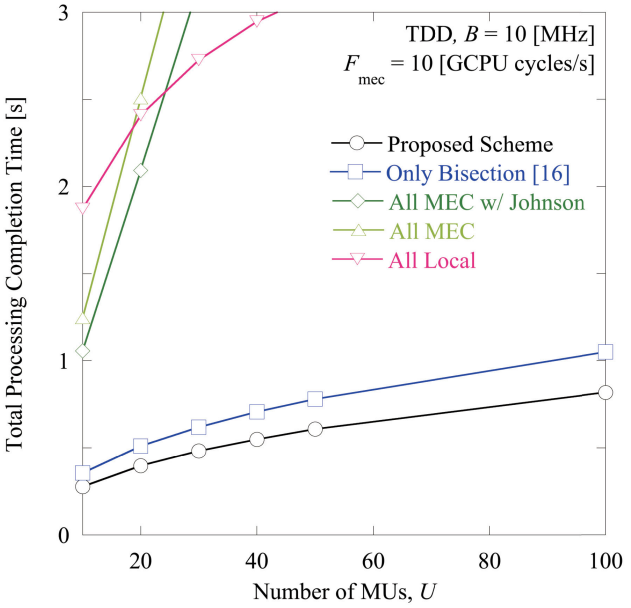


FIGURE 8. Impact of the number of MUs, U.

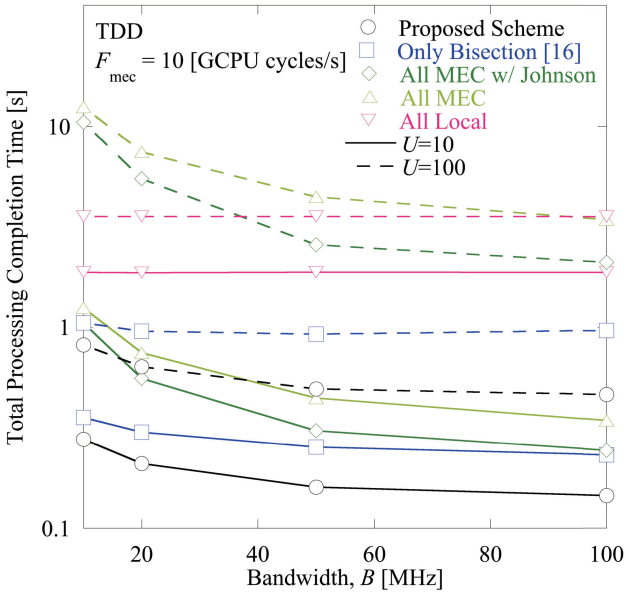


FIGURE 9. Impact of bandwidth, B [MHz].

$U = 10, 100$ and $F_{\text{mec}} = 10$ [GCPU cycles/s] as a function of bandwidth B [MHz]. When B is small, the time taken for offloading increases due to the small bandwidth, indicating that the effect of using MEC server is small. When the number of MUs $U = 100$ and the bandwidth $B = 100$ [MHz], the proposed scheme reduces total processing completion time by about 48% compared to the conventional scheme. By increasing B, the effect of offloading to the MEC server appears; the proposed scheme can significantly reduce the total processing completion time compared to the conventional scheme. As a result, it is found that there is a need to secure sufficient bandwidth when the MEC server is used.

C. THE IMPACT OF DUPLEXING ON PROPOSED SCHEME

We evaluate the impact of the duplex technique on the proposed scheme. In this paper, we consider the following two duplex schemes.

1) TIME DIVISION DUPLEX (TDD)

In TDD, since the same frequency band is assigned to UL and DL, there are restrictions on the timing of UL and DL. Thus, as shown in Fig. 10(a), even if the computation of a task is completed at MEC, it cannot be returned by DL until all MUs complete UL transmission.

2) FREQUENCY DIVISION DUPLEX (FDD)

In FDD, since two different frequency bands are assigned to UL and DL, UL and DL transmissions can be carried out at the same time. Thus, as shown in Fig. 10(a), even if UL transmission of all MUs is not yet completed, computing results from the MEC server can be returned by DL once its computation is completed. For FDD, let $\eta \in [0, 1]$ denote the fraction of the bandwidth allocated to UL. The channel

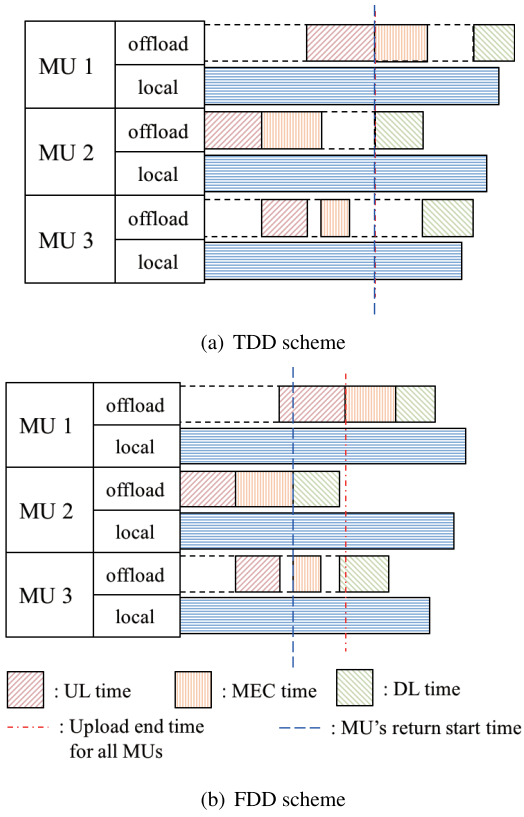


FIGURE 10. Two types of duplex scheme.

capacity is given by

$$C_{\text{UL},u} = \eta \times B \times \max\{\log_2(1 + \text{SINR}_u^{\text{UL}}),\ R_{\max}^{\text{UL}}\}, \quad (33)$$

$$C_{\text{DL},u} = (1-\eta) \times B \times \max\{\log_2(1+\text{SINR}_u^{\text{DL}}),\ R_{\max}^{\text{DL}}\}. \quad (34)$$

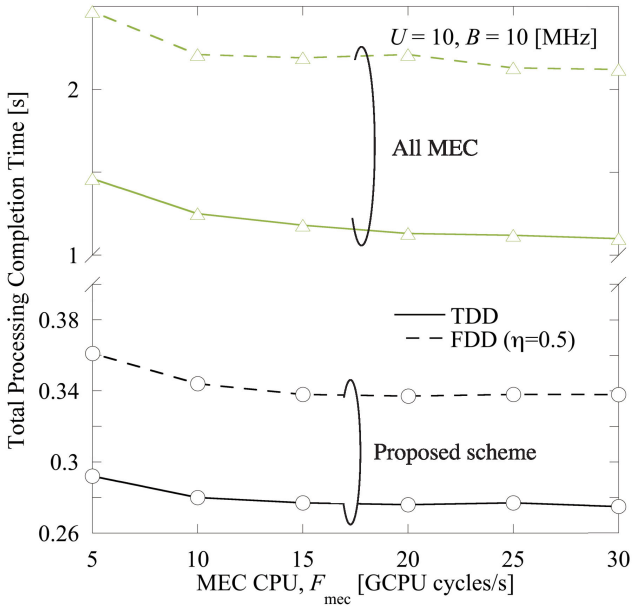


FIGURE 11. Comparison of average total processing completion time in each duplex scheme and the impact of MEC CPU capability F_{mec}.

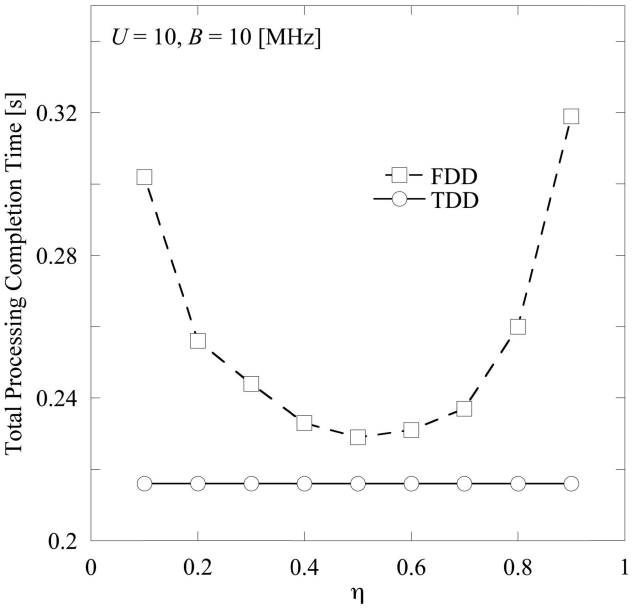


FIGURE 12. Comparison of average total processing completion time when bandwidth is changed in FDD.

where

$$\text{SINR}_u^{\text{UL}} = \frac{P_{\text{MU}} h_u}{P_{\text{MU}} \frac{\sum_{U'} h_{u'}}{U} + \eta \times BN_0}, \tag{35}$$

$$\text{SINR}_u^{\text{DL}} = \frac{P_{\text{BS}} h_u}{P_{\text{BS}} \sum_{\mathcal{K}} h_{k,u} + (1-\eta) \times BN_0}. \tag{36}$$

Fig. 11 shows the average total processing completion time in the case of using TDD and FDD with $\eta = 0.5$ as a function of F_{mec} for $U = 10$ and $B = 10$ [MHz]. For comparison, the performances when offloading all computation tasks are also shown. As shown in Fig. 11, the total processing completion time can be reduced as F_{mec} increases. Furthermore, the schemes with TDD provides shorter total processing completion time. The reason for this can be explained as follows. In FDD, although the computing result can be returned immediately after completion of computing MEC server, the allocated bandwidth for UL/DL is half of TDD. Thus, FDD requires more time for UL/DL time than TDD.

Fig. 12 shows the average total processing completion time with η as a parameter for $U = 10$, $B = 10$ [MHz], and $F_{\text{mec}} = 10$ [GCPU cycles/s]. As shown in Fig. 12, setting $\eta = 0.5$ provides the minimum total processing completion time for the proposed scheme with FDD. However, the proposed scheme with TDD provides the smaller total processing completion time.

D. PERFORMANCE GAP BETWEEN PROPOSED SCHEME AND OPTIMAL SCHEME

Finally, let us show the performance gap between the proposed scheme and the exhaustive search based optimal solution has been evaluated. The complexity for exhaustive search is cumbersome even for a small number of users such as 4,

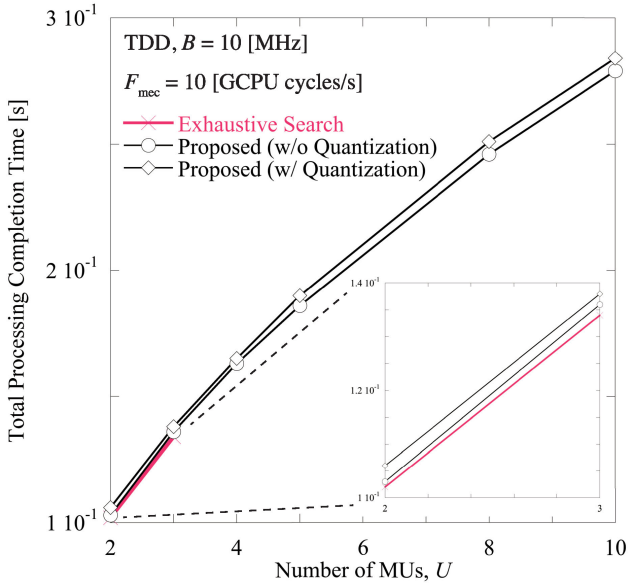


FIGURE 13. Performance comparison between the proposed scheme and the exhaustive search.

since the search space is infinite, i.e., the resolution for task split can be set to be small without any bound. Thus, we evaluate the performance gap only for a small number of users, i.e., $U = 2$ and 3. For exhaustive search, $D_{\text{res}} = 1$ [kbit] resolution is considered. In order for fair comparison, the performance of the proposed scheme with the same resolution $D_{\text{res}} = 1$ [kbit] is plotted (labeled as "Proposed scheme (w/ Quantization)"). In this case, the solutions with the offload size $l_{\text{UL},u}$ obtained by the bisection search is round down to $D_{\text{res}} \times \lfloor l_{\text{UL},u}/D_{\text{res}} \rfloor$ where $\lfloor x \rfloor$ returns the largest integer small than or equal to x. The total processing completion time is plotted as a function of number of users U in Fig. 13.

As it can be seen from the figure that the proposed algorithm can achieve the-close-to-optimal solution. The slight performance degradation is due to the decoupling of time allocation and clock allocation in original problem **(P)** into two subproblems **(P1)** and **(P2)**. However, the proposed scheme exhibits much lower computational complexity compared to the exhaustive search for solving original problem **(P)**.

VI. CONCLUSION

In this paper, for minimizing the total processing completion time of all the tasks, we have proposed a radio resource and computing resource allocation scheme. The scenario, when multiple MUs have tasks that can be divided into local tasks and offload tasks, was considered. Different from the prior works, the impact of a downlink transmission from BS to MUs, which is not negligible anymore in certain application types, was also taken into account. We first formulated the optimization problem to minimize the total processing completion time of all tasks. For a mixed discrete-continuous optimization problem, we proposed a two-step sub-optimal radio and computing resources allocation scheme which iteratively performs the bisection search method and Johnson's algorithm. The numerical results showed that the proposed scheme could reduce the total processing completion time by about 25% on average when the number of MUs is $U = 100$ compared with the conventional scheme. In this paper, MUs are assumed to be static. The scheduling for high mobility MU is left as an interesting future study.

APPENDIX A

PROOF OF THE FEASIBILITY CONDITION

The feasibility condition (22), (23), and (24) will be derived following [16]. However, unlike [16], this paper considers DL time in addition to UL time and computing time and allocates resources on the MEC server on a time-division manner.

First, since it is necessary for (19) to hold for all MUs, one of feasibility conditions is

$$\frac{A_u l_{\mathrm{UL},u}^{\min}}{F_{\mathrm{mec}}} \leq T^{(\mathsf{P1})}. \tag{37}$$

Next, rewrite (18) using a_u, b_u, and c_u as

$$\frac{a_u}{x_u} + b_u + \frac{c_u}{y_u} \leq T^{(\mathsf{P1})}. \tag{38}$$

Thus, we have

$$x_u \geq \frac{a_u}{\left(T^{(\mathsf{P1})} - b_u\right)}, \tag{39}$$

$$y_u \geq \frac{c_u x_u}{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u}. \tag{40}$$

Here, since the following equation must be established from (39) and (20), one of the feasibility conditions is

$$\sum_{u=1}^{U} \frac{a_u}{\left(T^{(\mathsf{P1})} - b_u\right)} \leq 1. \tag{41}$$

In addition, combining Eqs. (40) and (21) gives

$$\sum_{u=1}^{U} \frac{c_u x_u}{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u} \leq 1. \tag{42}$$

Equation (42) is satisfied only when the minimum value of $\sum_{u=1}^{U} \frac{c_u x_u}{(T^{(\mathsf{P1})} - b_u) x_u - a_u}$ does not exceed 1, and the constraint conditions at this time are $\sum_{u=1}^{U} x_u = 1$ and $x_u \geq \frac{a_u}{(T^{(\mathsf{P1})} - b_u)}$. Therefore, it is necessary to solve the following optimization problem:

$$\min_{\{x_u\}} \quad \sum_{u=1}^{U} \frac{c_u x_u}{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u} \tag{P1B}$$

$$\text{subject to} \quad \sum_{u=1}^{U} x_u = 1, \quad x_u \geq \frac{a_u}{\left(T^{(\mathsf{P1})} - b_u\right)}. \tag{43}$$

Here, function $\frac{cx}{(T^{(\mathsf{P1})} - b)x - a}$ is a convex function for $x > \frac{a}{(T^{(\mathsf{P1})} - b)}$ with $a \geq 0, T^{(\mathsf{P1})} > 0, b \geq 0$, and $c \geq 0$. Therefore, the optimization problem **(P1B)** is a convex optimization problem having one linear constraint, and using the corresponding Lagrangian multiplier method, it is represented as

$$\mathcal{L}\left(\{x_u\}, \lambda\right) = \sum_{u=1}^{U} \frac{c_u x_u}{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u} + \lambda \left(\sum_{u=1}^{U} x_u - 1 \right). \tag{44}$$

Partial differentiation is performed on each variable with respect to (44) and we obtain

$$\frac{\partial \mathcal{L}}{\partial x_u} = -\frac{a_u c_u}{\left\{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u\right\}^2} + \lambda, \quad \forall u \in \mathcal{U}, \tag{45}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{u=1}^{U} x_u - 1. \tag{46}$$

From the above (45) and (46), we obtain

$$x_u^* = \frac{a_u + \sqrt{\frac{a_u c_u}{\lambda}}}{T^{(\mathsf{P1})} - b_u}, \tag{47}$$

$$\sqrt{\lambda} = \frac{\sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathsf{P1})} - b_u}}{1 - \sum_{u=1}^{U} \frac{a_u}{T^{(\mathsf{P1})} - b_u}}. \tag{48}$$

Substituting x_u^* given by (47) into the objective function of the optimization problem **(P1B)** and letting $\sqrt{\lambda}$ be (48), we obtain minimum value

$$\begin{aligned} \text{min value} &= \sum_{u=1}^{U} \frac{c_u x_u}{\left(T^{(\mathsf{P1})} - b_u\right) x_u - a_u} \\ &= \frac{\sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathsf{P1})} - b_u}}{1 - \sum_{u=1}^{U} \frac{a_u}{T^{(\mathsf{P1})} - b_u}} \sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathsf{P1})} - b_u} \\ &\quad + \sum_{u=1}^{U} \frac{c_u}{T^{(\mathsf{P1})} - b_u}. \end{aligned} \tag{49}$$

Equation (42) is satisfied only if the minimum value of $\sum_{u=1}^{U} \frac{c_u x_u}{(T^{(\mathsf{P1})} - b_u) x_u - a_u}$ with constraint conditions

$\sum_{u=1}^{U} x_u = 1$ and $x_u \geq \frac{a_u}{(T^{(\mathrm{P1})}-b_u)}$ does not exceed 1, so this minimum value should not exceed 1. Therefore, it can be expressed by

$$\frac{\sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathrm{P1})}-b_u}}{1-\sum_{u=1}^{U} \frac{a_u}{T^{(\mathrm{P1})}-b_u}} \sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathrm{P1})}-b_u} + \sum_{u=1}^{U} \frac{c_u}{T^{(\mathrm{P1})}-b_u} \leq 1. \quad (50)$$

Thus, from (50), the further feasibility condition is expressed by

$$\left(\sum_{u=1}^{U} \frac{\sqrt{a_u c_u}}{T^{(\mathrm{P1})}-b_u}\right)^2 \leq \left(1-\sum_{u=1}^{U} \frac{a_u}{T^{(\mathrm{P1})}-b_u}\right) \times \left(1-\sum_{u=1}^{U} \frac{c_u}{T^{(\mathrm{P1})}-b_u}\right). \quad (51)$$

$$\sum_{u=1}^{U} \frac{a_u}{T^{(\mathrm{P1})}-b_u} \leq 1, \quad \sum_{u=1}^{U} \frac{c_u}{T^{(\mathrm{P1})}-b_u} \leq 1. \quad (52)$$

APPENDIX B
JOHNSON'S ALGORITHM

In this appendix, a brief example of the Johnson's algorithm is provided. Let us consider the scenario where four jobs $\{J_1, J_2, J_3, J_4\}$ are processed by three machines X, Y, and Z in the order of X $\rightarrow$ Y $\rightarrow$ Z. Let X_i, Y_i, and Z_i denote the processing time of job J_i at each machine. Table 5 shows the processing time of each job required at each machine.

For three machine scheduling problem, two *virtual machines* are created. The first virtual machine α is by combining two machines X and Y. The second virtual machine β is by combining two machines Y and Z. For this case, we have the following theorem [25, Theorem 2].

Theorem 2: If $\min X_i \geq \max Y_i$, $1 \leq i \leq U$, then the optimal three stage scheduling is given by the following rule. Job J_i precedes job J_j if the following condition is satisfied

$$\min\left(X_i + Y_i, Z_j + Y_j\right) < \min\left(X_j + Y_j, Z_i + Y_i\right). \quad (53)$$

Since in our problem, machine X and machine Y correspond to uploading and processing at MEC, the condition $\min X_i \geq \max Y_i$, $1 \leq i \leq U$ can be satisfied with high probability. Based on Theorem 2, the following simple scheduling algorithm is obtained.

1) The minimum processing time is searched from the processing time of jobs.
2) Job scheduling.
   - If the searched processing time is for virtual machine α, then the corresponding job is scheduled first.
   - If it is for virtual machine β, then the corresponding job is scheduled last.
3) Remove the processing time of the job scheduled in step 2 from both virtual machines.
4) The same procedure is repeated until all the jobs are scheduled.

TABLE 5. Processing Time on each machine.

Job \ Machine	Processing time [s]		
	X_i	Y_i	Z_i
J_1	6	1	4
J_2	8	4	4
J_3	2	2	7
J_4	8	3	6

TABLE 6. Processing time on each virtual machine.

Job \ Virtual Machine	Processing time [s]	
	$X_i + Y_i$	$Z_i + Y_i$
J_1	7	5
J_2	12	8
J_3	4	9
J_4	11	9

Processing order			
J_3			
↓			
J_3			J_1
↓			
J_3		J_2	J_1
↓			
J_3	J_4	J_2	J_1

Let us consider the scenario for the jobs whose processing time are shown in Table 6. Firstly, the minimum processing time is searched from the processing time of the jobs shown in Table 6. In this example, the minimum processing time in Table 6 is 4 of job J_3, which is the processing time of α. Since this processing time is required for virtual machine α, the corresponding job J_3 is scheduled first and removed from the list. Then the minimum processing time is searched in Table 6 with J_3 being removed. Then, job J_1 whose processing time 5 is minimum is selected. Since the corresponding minimum time is for virtual machine β, it is scheduled last and removed from the table. The same procedure is repeated until all the jobs are scheduled.

REFERENCES

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[2] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[3] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE ICASSP*, Shanghai, China, Mar. 2016, pp. 3516–3520.

[4] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 14–22, Jun. 2013.

[5] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

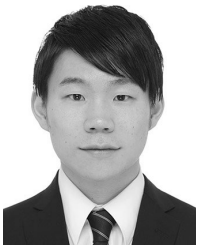
[6] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Halmstad, Sweden, Sep. 2016, pp. 288–294.

[7] J. Zhou, F. Wu, K. Zhang, Y. Mao, and S. Leng, "Joint optimization of offloading and resource allocation in vehicular networks with mobile edge computing," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Hangzhou, China, Oct. 2018, pp. 1–6.

[8] M. Li, F. R. Yu, P. Si, H. Yao, E. Sun, and Y. Zhang, "Energy-efficient M2M communications with mobile edge computing in virtualized cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[9] X. Lin, H. Zhang, H. Ji, and V. C. M. Leung, "Joint computation and communication resource allocation in mobile-edge cloud computing networks," in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Beijing, China, Sep. 2016, pp. 166–171.

[10] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[11] T. Zhao, S. Zhou, X. Guo, and Z. Niu, "Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.
[12] H. Yu, Q. Wang, and S. Guo, "Energy-efficient task offloading and resource scheduling for mobile edge computing," in *Proc. IEEE Int. Conf. Netw., Architecture Storage (NAS)*, Chongqing, China, Oct. 2018, pp. 1–4.
[13] W. Labidi, M. Sarkiss, and M. Kamoun, "Energy-optimal resource scheduling and computation offloading in small cell networks," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, Sydney, NSW, Australia, Apr. 2015, pp. 313–318.
[14] X. Zhang, Y. Mao, J. Zhang, and K. B. Letaief, "Multi-objective resource allocation for mobile edge computing systems," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–5.
[15] 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, USA, Dec. 2016, pp. 1–6.
[16] H. Q. Le, H. Al-Shatri, and A. Klein, "Efficient resource allocation in mobile-edge computation offloading: Completion time minimization" in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 2513–2517.
[17] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-user mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–7.
[18] 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.
[19] J. Zhang, W. Xia, Z. Cheng, Q. Zou, B. Huang, F. Shen, F. Yan, and L. Shen, "An evolutionary game for joint wireless and cloud resource allocation in mobile edge computing," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Oct. 2017, pp. 1–6.
[20] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.
[21] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.
[22] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE J. Sel. Topics Signal Process*, vol. 13, no. 3, pp. 392–407, Jun. 2019.
[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[24] J. Blazewicz, K. H. Ecker, G. Schmidt, and J. Weglarz, *Scheduling in Computer and Manufacturing Systems*. Berlin, Germany: Springer-Verlag, 1994.
[25] S. M. Johnson, "Optimal two- and three-stage production schedules with setup times included," *Naval Res. Logistics Quart.*, vol. 1, no. 1, pp. 61–68, 1954.
[26] *LTE; Evolved Universal Terrestrial Radio Access (EUTRA); Radio Frequency (RF) Requirements for LTE Pico Node B (3GPP TR 36.931 version 9.0.0 Release 9)*, document ETSI TR 136 931, Version 9.0.0, ETSI, May 2011, pp. 10–11.



RYUJI KOBAYASHI received the B.E. degree in engineering from Tokyo City University, Tokyo, Japan, in 2017, and the M.E. degree from The University of Electro-Communications, Tokyo, in 2019. His current research interest includes resource allocation.

KOICHI ADACHI (S'06–M'09) received the B.E., M.E., and Ph.D. degrees in engineering from Keio University, Japan, in 2005, 2007, and 2009, respectively. From 2007 to 2010, he was a Research Fellow with the Japan Society for the Promotion of Science (JSPS). He was a Visiting Researcher with the City University of Hong Kong, in April 2009, and a Visiting Research Fellow with the University of Kent, from June to August 2009. From May 2010 to May 2016, he was with the Institute for Infocomm Research, A*STAR, in Singapore. He is currently an Associate Professor with The University of Electro-Communications, Japan. His research interests include cooperative communications and energy efficient communication technologies.

He was a recipient of the Excellent Editor Award from the IEEE ComSoc MMTC, in 2013. He has served as the General Co-Chair for the 10th and the 11th IEEE Vehicular Technology Society Asia Pacific Wireless Communications Symposium (APWCS), the Track Co-Chair for the Transmission Technologies and Communication Theory of the 78th and 80th IEEE Vehicular Technology Conference, from 2013 to 2014, and the Symposium Co-Chair of the Communication Theory Symposium of the IEEE GLOBECOM 2018. He was an Associate Editor of the *IET Transaction on Communications*, from 2015 to 2017, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2016 to 2018. He has been an Associate Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS, since 2016. He has served as an Exemplary Reviewer for the IEEE COMMUNICATIONS LETTERS, in 2012, and the IEEE WIRELESS COMMUNICATIONS LETTERS, in 2012, 2013, 2014, and 2015.