IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Energy Efficient Communication and Computation Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond

YAN KYAW TUN[1], DO HYEON KIM[1], MADYAN ALSENWI[1], NGUYEN H. TRAN[2], (Senior Member, IEEE), ZHU HAN[1,3], (Fellow, IEEE), AND CHOONG SEON HONG[1], (Senior member, IEE)

[1]Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea
[2]School of Computer Science, The University of Sydney, NSW 2006, Australia
[3]Electrical and Computer Engineering Department, University of Houston, Houston, TX 77004 USA

Corresponding author: Choong Seon Hong (e-mail: cshong@khu.ac.kr).

This research was partially supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2020-2015-0-00742) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation), and supported by IITP grant funded by the MSIT (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing).

ABSTRACT Multi-access edge computing (MEC) enables mobile users to offload their computation tasks to the server located at the edge of the cellular network. Thereby, MEC prolongs the battery lifespan of mobile devices and significantly enhances their computation capacities. However, a significant challenge is to offload the computation tasks to the MEC server in an energy-efficient manner. Meanwhile, in the fifth-generation (5G) networks, mobile users with different service requirements classified as enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (URLLC) users will coexist in the current cellular network. Therefore, it is important to appropriately multiplex these users in the cellular network. In this work, we address the issues of these two promising technologies together. Firstly, we formulate an energy-efficient task offloading, and scheduling of eMBB and URLLC users as a mixed-integer non-linear problem. Then, we decompose the problem into multiple sub-problems in order to transform into convex form and alternately solve them until converging to the desired solutions by using the block coordinate descent (BCD) algorithm. Finally, we demonstrate numerical results to prove the superior effectiveness in the performance of our proposed algorithm over classical existing schemes.

INDEX TERMS Multi-access edge computing (MEC), resource slicing, eMBB-URLLC coexistence, block coordinate descent (BCD).

I. INTRODUCTION

WITH the explosive growth of the Internet of Things (IoT) devices, computation-intensive applications (e.g., Augmented Reality (AR), face recognition, Virtual Reality (VR), online gaming, and traffic monitoring) are appearing as an integral part of our daily activities. However, as the computation capacity (i.e., central processing unit (CPU) capacity) and battery lifetime of IoT devices are limited, processing data on the local device becomes a challenging issue. One of the promising solutions to address the above argument is the mobile cloud computing (MCC) [1], [2]. In MCC, the energy consumption at the resource-constrained devices can be reduced by offloading its computation-intensive tasks to the cloud server via a cellular network. Then, the cloud server executes offloaded tasks and sends the output back to the devices. However, the cloud server is far from the mobile devices, and the delay experienced by the services at the devices can be significantly increased. Recently, the wireless communication industry and academic research community have introduced the new technology called multi-access edge computing (MEC) [3]. In MEC, a server is deployed at the edge of the radio access network, e.g., at the small cell, macro

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

base station, relay station, and an access point, and provides computation services to the mobile devices [4]. Compared with MCC, MEC can reduce the delay of the mobile devices as the computing resource is nearer to the devices than the cloud server. Notably, computation capacity of a MEC server is limited when compared with the cloud server; therefore, efficient resource allocation in MEC system comes up as a challenge. To address this issue, several current research works have discussed an efficient computation-intensive task offloading and resource allocation algorithms in MEC [5]–[8].

According to the 3GPP latest release, multiple communication services such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low latency communication (URLLC) will be provided in 5G. To summarize, eMBB service is the advancement of the existing mobile broadband service in the current LTE networks with higher throughput for delay-sensitive applications. URLLC is the new service for highly reliable and low latency applications, e.g., self-driving car, industrial automotive, and remote surgery, and mMTC is for IoT devices employed in smart city. Though new wireless services will be available in 5G, there remains several challenging issues to be addressed. Amongst them, one crucial challenge is to schedule services with different QoS requirements in the existing cellular network. In the previous works, authors addressed the scheduling of eMBB and URLLC traffic in 5G new radio access networks [9]–[12].

A. RESEARCH CONTRIBUTIONS

All of the existing works have discussed an efficient task offloading, and computation resource allocation mechanism in the MEC system and further on the scheduling of the eMBB and URLLC users separately. However, these two research issues are coupled because eMBB users will need high computation service due to their limited computation capacity (i.e., CPU resources) and battery lifetime. In this work, different from the previous works, we jointly consider an energy-efficient task offloading, and multiplexing of the URLLC and eMBB users. In summary, the contributions of this work can be expressed as follows:

- We formulate an efficient joint task offloading, and scheduling of eMBB and URLLC users problem that minimizes energy consumption and maximizes the achievable data rate of the eMBB users subject to the latency of eMBB users, the CPU capacity of the MEC server, the maximum transmit power of eMBB users, the reliability of the URLLC traffic, and the resource block allocation to the eMBB users constraints. The above optimization problem is a non-convex and NP-hard problem, which has the combinatorial complexity.
- To solve the proposed problem, we first relax the binary variable and decompose the problem into multiple sub-problems. Then, an iterative algorithm, block coordinate descent (BCD), is applied to solve the relaxed problem.
- In the simulation section, firstly, the convergence rate of our proposed algorithm is demonstrated. Then, we compare the performance of our proposed algorithm with other schemes: equal resource sharing, all local computing, and all offloading schemes. There, our proposed scheme achieves a significant performance gain.

The remainder of this paper is organized as follows: Section III describes the related works and Section III presents the system model and problem formulation. The proposed solution approach for the formulated problem is introduced in Section IV and Section V discusses the simulation results. Finally, Section VI concludes the paper.

II. RELATED WORKS

The existing works can be categorized into two groups: i) energy efficient task offloading in the multi-access edge computing, and ii) resource slicing for eMBB and URLLC traffic.

(i) Energy-efficient task offloading in multi-access edge computing: The work of [13] introduced a wireless powered-enabled multi-user MEC system where the wireless access point integrated with multiple antennas and MEC server is installed to provide wireless power and computation resources to the mobile users in its coverage area. In [14], authors have proposed energy efficient task offloading problem in the mobile edge computing-enabled radio access network where they considered multiple base stations scenario with the consideration of the inter-cell interference. Then, they developed an artificial fish swarm algorithm-based scheme to solve the proposed problem. In [15], the authors introduced the communication-computation tradeoff study in the multi-server MEC system and proposed an optimal offloading policy where each user can offload its computation tasks to multiple servers at the same time. The work of [16] considered the performance guaranteed computation offloading in multi-cell networks where they tried to minimize the energy consumption of the mobile users in the network. In [17], authors introduced an efficient online algorithm in order to make joint base station sleeping and computation task offloading decisions in the ultra dense cellular networks. In this work, authors assumed that the mobile edge computing servers are deployed at the macro base stations to which mobile users offloaded their computation tasks. However, authors did not take into account the inter-cell interference between the base stations. The work in [18] proposed game theory-based data offloading from the mobile devices (MDs) to the MEC servers where authors deployed coalitions to manage MDs data offloading as well as to demonstrate the relationship between MDs and MEC servers. Then, the pricing scheme was applied to stimulate the MDs for data offloading. In [19], the authors proposed an auction-based resource allocation problem in the hierarchical mobile edge computing where they tired to maximize the profit of the service providers. Moreover, the work in [20] proposed the radio and computation resource allocation problem in the multi-access edge computing system. Then, the authors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

decomposed the proposed problem into upper and lower problems. Finally, they exploited the potential game and evolutionary game in order to address both upper-level and lower-level problems with the aim of minimizing the cost of the mobile users and maximizing the profit of the MEC servers. In [21], the authors proposed a hierarchical system of three layers to address the problems facing in the fog computing networks. Then, they implemented the proposed system as the Stackelberg subgame for the interaction between authorized data service subscribers (ADSSs) and data service providers (DSOs), moral hazard modeling for the interaction between the fog nodes (FNs) and DSOs, and the student project allocation matching game for the interaction between FNs and ADSSs. The work of [22] introduced a distributed computation task offloading and radio resource allocation problem in the mobile edge computing system. In that work, authors considered the multi-cell scenario and the inter-cell interference was also taken into account. Then, they applied the matching game in order to solve the proposed problem.

(ii) Resource slicing of eMBB and URLLC traffic: The work of [23] studied the efficiency of both orthogonal and non-orthogonal multiple access in the uplink of a multi-cell Cloud Radio Access Network (C-RAN) system for multiplexing of eMBB and URLLC users. In [10], the authors studied the joint eMBB and URLLC traffic scheduling problem with the aim of maximizing the achievable data rate of the eMBB users while satisfying the reliability constraint of the URLLC traffic. Moreover, the authors in [24] studied uplink communication resource slicing for eMBB and grant-free URLLC traffic. The work of [25] proposed a RAN slicing problem that enables synchronized multi-point (CoMP) transmissions for multicast eMBB and bursty URLLC service as a multi-time optimization problem. Then, the proposed multi-time scale problem is transformed into the multiple single-time problems and the authors applied the iterative algorithm to address the transformed problem. In [26], the authors proposed the problem of downlink multiplexing of eMBB and URLLC traffic in the 5G networks, then, an indicator-free approach is applied to solve the problem. The authors in [27] proposed the deep reinforcement learning based radio resource slicing for eMBB and URLLC traffic. The work of [28] proposed a resource scheduling framework for eMBB and URLLC based on the puncturing approach. They leveraged the Deep Reinforcement Learning (DRL) to find the number of punctured mini-slots from all eMBB users. Then, [29] proposed the use cases of the URLLC traffic in the 5G new radio and the authors in [30] proposed the AI-enabled radio resource slicing for eMBB and URLLC users.

In this paper, we find the critical requirements frequently overlooked in the previous works, and formulate the joint efficient computation task offloading and scheduling of eMBB and URLLC traffic. In addition, we employ the block-coordinate descent (BCD) algorithm in order to solve the proposed optimization problem.

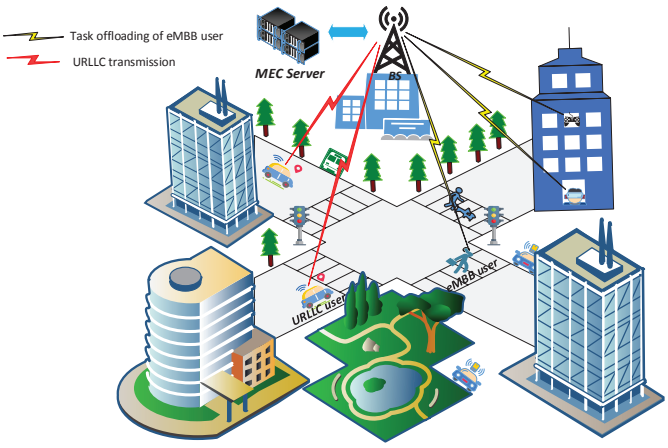


FIGURE 1: System Model.

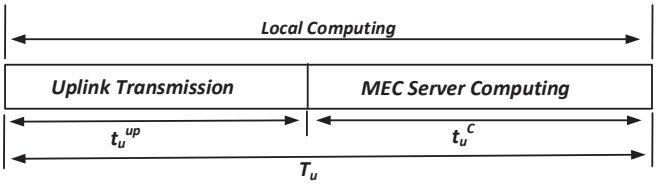


FIGURE 2: Communication and computation latency.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

As shown in Fig. 1, we focus on the multiuser mobile edge computing system with a single base station (BS) attached with the MEC server and eMBB users (i.e., AR and VR) in the set $\{1, 2, \ldots, U\}$ defined by $\mathcal{U}$. The BS is operating on the system bandwidth F and is orthogonally sliced [31] into two static portions, such as F_b for eMBB users and $F_s = F - F_b$ for the traffic of URLLC users. However, in this work, dynamic resource slicing is studied, where we consider the scenario with high traffic of URLLC users. In other words, the portion of available resource F_s is not sufficient to provide services to all URLLC users. In such a case, the BS will puncture the resource allocated to the eMBB users, meanwhile ensuring the QoS requirement of each eMBB user. The fraction of the system bandwidth for eMBB users F_b is divided into a set of resource blocks (RBs) $\mathcal{B} = \{1, 2, \ldots, B\}$, where each RB has bandwidth δ. Moreover, the time duration of each resource block is one millisecond. Here, we consider the scenario where eMBB users are required to compute different computation tasks under different latency constraints. In this multiuser MEC network, each eMBB user has a computation task defined as $\gamma_u = \{d_u, c_u, T_u\}$, where d_u is the total data size of the input data of the eMBB user u, c_u denotes the total number of CPU cycles required to accomplish one bit of the computation task, and T_u is the maximum tolerable latency or the execution deadline of the task of user u. We assume that the BS has complete information on the channel state

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

TABLE 1: Summary of Key Notations.

Notation	Definition
$\mathcal{U}$	Set of eMBB users, $\|\mathcal{U}\| = U$
F	Total system bandwidth
F_b	Fraction of system bandwidth allocated to eMBB users
F_s	Fraction of system bandwidth allocated to URLLC users
$\mathcal{B}$	Set of resource blocks, $\|\mathcal{B}\| = B$
d_u	Total input data size of eMBB user u
c_u	Required CPU cycles to accomplish one bit of the input data of eMBB user u
T_u	The execution deadline of the task of eMBB user u
l_u	The offloaded data size of the task of eMBB user u
t_u^L	The local computation execution time of eMBB user u
E_u^L	The local computation energy of eMBB user u
y_u^b	Resource block assignment variable
M	Number of minislots divided in each resource block
L_m	Traffic of URLLC users at minislot m
$L_{\mathbf{max}}$	Maximum traffic of URLLC users that can be served at a time slot
w_u	Weight of puncturing eMBB user u
g_u^b	Achievable channel gain of eMBB user u
P_u^b	Transmit power of eMBB user u
$R_{u,b}$	Achievable data rate of eMBB user u on resource block b
t_u^{up}	The uplink transmission delay experienced by eMBB user u
f^C	The total CPU capacity of the MEC server
f_u^C	The CPU capacity of the MEC server that is allocated to eMBB user u
E_u^{Off}	The energy consumption of eMBB user u for offloading data
R_s	The achievable data rate of URLLC user s
P_s	The transmit power of URLLC user s
g_s	The achievable channel gain of URLLC user s
R_{urllc}	The total achievable data rate of URLLC users

information (CSI), local energy consumption, and required CPU cycles to execute one bit of data. We also assume that the size of the input data of all eMBB users can be obtained by the feedback signal [5]. For each eMBB user, its task can be executed locally or offloaded to the MEC server, and then the server executes it. In this work, we consider the partial task offloading scenario where the total input data size of each user is divided into remote computing and local computing as l_u and $(d_u - l_u)$, respectively.

1) Local computing: Let us define the maximum computation capacity (CPU cycles per second) of eMBB user u as f_u^l. If user u executes a portion of its task locally, the task execution time (i.e., latency) to accomplish the task is as follows:

$$t_u^L = \frac{c_u(d_u - l_u)}{f_u^l}, \forall u \in \mathcal{U}. \tag{1}$$

Then, the energy consumption at eMBB user u for local computing can be expressed as follows:

$$E_u^L = k(f_u^l)^2 c_u(d_u - l_u), \forall u \in \mathcal{U}, \tag{2}$$

where k is the system constant which depends on the chip architecture of the eMBB users' device, and set as $k = 5 \times 10^{-27}$.

2) Mobile Edge Computing: When eMBB users offload input data to the MEC server, the BS will allocate resource blocks to the eMBB users. Let us introduce a new variable $y_u^b \in \{0, 1\}$ as the resource block assignment variable, which indicates whether or not resource block b is assigned to eMBB user u:

$$y_u^b = \begin{cases} 1, & \text{if eMBB user } u \text{ is assgined to resource} \\ & \text{block } b\text{ ,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here, the minislots produced through dividing each resource block of the time slot are represented as M. Meanwhile, the traffic of URLLC users may happen within the time slot of which the RBs are already allocated to the eMBB users. Due to the hard latency requirements, the traffic of URLLC users cannot be postponed to the next time slot. Hence, the arriving traffic of URLLC users is scheduled in order to transmit in the next minislot and the zero transmit power for the overlapped eMBB users. Let us consider L_m as the variable that represents the traffic of URLLC users, which comes at a minislot, denoted by m. L_m can be represented as a Bernoulli distribution with success probability p (i.e. the likelihood of URLLC transmission at a minislot).

There is a direct relation between the loss rate of user data of eMBB within each time slot and the punctured resources of eMBB user [10]. Let w_u define the weight of puncturing eMBB user u, i.e., $w_u \in [0, 1]$, where $u \in \mathcal{U}$ and $\boldsymbol{w} = (w_1, w_2, \ldots, w_U)$. The higher the w_u value, the higher the probability of eMBB users being punctured by the traffic of URLLC users. Therefore, the achievable data rate of eMBB user u on resource block b with puncturing weight w_u is [9]:

$$R_{u,b} = \delta\left(1 - w_u \times \frac{L}{L_{\mathbf{max}}}\right)\log_2\left(1 + \frac{P_u^b g_u^b}{N_0}\right), \quad \forall b \in \mathcal{B}, \forall u \in \mathcal{U}, \tag{4}$$

where $L = \sum_{m \in M} L_m$ follows the binomial distribution with parameter M and p. The maximum traffic of URLLC users that can be served at each time slot is represented by $L_{\mathbf{max}}$, P_u^b and g_u^b are the uplink transmit power and achieved channel gain of eMBB user u on resource block b, respectively, and N_0 represents the additive white Gaussian noise power. The term $w_u \times \frac{L}{L_{\mathbf{max}}}$ is the estimation of the punctured resources of eMBB user u by the traffic of URLLC users. Then, the total achievable instantaneous data rate of eMBB user u is as follows:

$$R_u = \sum_{b=1}^{B} y_u^b R_{u,b}, \forall u \in \mathcal{U}. \tag{5}$$

The uplink transmission delay/time of eMBB user u to the MEC server is:

$$t_u^{\text{up}} = \frac{l_u}{R_u}, \forall u \in \mathcal{U}. \tag{6}$$

Then, the total task execution time of eMBB user u, including the uplink transmission time (i.e., offloading time) and computation time at the MEC server can be expressed as:

$$t_u^{\text{Off}} = \frac{l_u}{R_u} + \frac{c_u l_u}{f_u^C}, \forall u \in \mathcal{U}, \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

where f_u^C is the CPU capacity of the MEC server that is allocated to eMBB user u. In this work, we consider weighted proportional allocation-based computation resource (i.e., CPU capacity) allocation [32]. Therefore, the CPU capacity of the MEC server allocated to eMBB user u is as follows:

$$f_u^C = \frac{l_u}{\sum_{u=1}^{U} l_u} f^C, \forall u \in \mathcal{U}, \tag{8}$$

where $\sum_{u=1}^{U} l_u$ is the total offloaded data of all eMBB users, and f^C is the CPU capacity of the MEC server at the BS. Finally, the energy consumption of eMBB user u for offloading data is calculated as:

$$E_u^{\text{Off}} = \sum_{b=1}^{B} y_u^b P_u^b \frac{l_u}{R_{u,b}}, \forall u \in \mathcal{U}. \tag{9}$$

Here, the energy consumption for downlink transmission is omitted because the data output from the MEC server is much smaller than the input data size.

The purpose of the URLLC scheduler is to get a URLLC placement weight vector $\boldsymbol{w}$ so that the arriving traffic of URLLC users can be scheduled. At the same time, the reliability of eMBB users transmission is considered. Depending on the formulation intended for maximizing the total average data rate of eMBB users, the URLLC placement weight vector calculation has an effect on eMBB users with a low data rate while protecting the users with the high data rate. For that reason, the risk on the eMBB transmission is considered by the URLLC scheduler in order to protect eMBB users with a low data rate. The distribution of the traffic of URLLC users among eMBB users by the calculated URLLC placement weight $\boldsymbol{w}$ considers the users with bad channel conditions. To incorporate this, this paper hereby describes conditional value-at-risk (CVaR) as a risk measure because of the fact that it captures the tail of the eMBB users' data rate. The CVaR supports the average of potential loss, which is more than the Value-at-Risk (VaR). β -VaR is the β -percentile distributed by random variable given as [33],

$$\mathsf{VaR}_\beta(R) = \mathsf{arg\ inf}_\nu\{\nu : P(R > \nu) \leq \beta\}, \tag{10}$$

where $R = \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_u^b$ and $\beta \in (0, 1)$. The CVaR function is defined as the expectation of the β function of the worst outcomes of R:

$$\mathsf{CVaR}_\beta(R) = \mathbb{E}[R|R > \mathsf{VaR}_\beta(R)]. \tag{11}$$

Furthermore, from [33]:

$$\zeta_\beta(R, \nu) := \nu + \frac{1}{1-\beta}\mathbb{E}[(R-\nu)^+], \tag{12}$$

and the CVaR_β of the random variable R is formulated as follows:

$$\mathsf{CVaR}_\beta(R) = \min_{\nu \in \mathbb{R}} \zeta_\beta(R, \nu). \tag{13}$$

Moreover, we rewrite $\mathbb{E}[R]$ as follows:

$$\mathbb{E}[R] = \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b \delta \left(1 - w_u \times \frac{\mathbb{E}[L]}{L_{\max}}\right) \log_2\left(1 + \frac{P_u^b g_u^b}{N_0}\right) \tag{14}$$

where $\mathbb{E}[L] = Mp$. Let us consider a set of URLLC users $\mathcal{S} = \{1, 2, \ldots, S\}$. In this work, we consider that the total incoming traffic of the URLLC users is random, and the number of URLLC users is a discrete finite set. Here, we consider that the total punctured resources to the eMBB users are equally divided among the URLLC users. Therefore, the achievable data rate of URLLC user s can be expressed as:

$$R_s = \sum_{u=1}^{U} \sum_{b=1}^{B} \frac{y_u^b \lambda_u}{S} \log_2\left(1 + \frac{P_s g_s}{N_0}\right), \forall s \in \mathcal{S}. \tag{15}$$

Then, the maximum achievable rate of URLLC users can be expressed as:

$$R_{urllc} = \sum_{s=1}^{S} \sum_{u=1}^{U} \sum_{b=1}^{B} \frac{y_u^b \lambda_u}{S} \log_2\left(1 + \frac{P_s g_s}{N_0}\right), \tag{16}$$

where $\lambda_u = (\delta \times w_u \frac{L}{L_{\max}})$, S is the total number of URLLC users, P_s and g_s are the transmit power and the achievable channel gain of URLLC user s, respectively. Then, the outage probability of URLLC traffic can be expressed as:

$$P(E) = Pr[R_{urllc} \leq L]. \tag{17}$$

B. PROBLEM FORMULATION

In this work, we firstly allocate physical resource blocks to the eMBB users who need to offload their computational tasks to the MEC server. Then, we propose the risk-sensitive approach to assign the resource to the incoming traffic of URLLC users, ensuring reliability constraint while minimizing the risk of the eMBB users who are offloading their tasks. Here, we want to minimize the overall energy consumption and maximize the achievable data rate of eMBB users considering the reliability of URLLC users. Therefore, we can write an optimization problem as:

$$\min_{\boldsymbol{y},\boldsymbol{l},\boldsymbol{w}} \left(\sum_{u=1}^{U} E_u^{\text{Off}} + \sum_{u=1}^{U} E_u^L\right) - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b} \tag{18}$$

$$\text{s.t. C1}: \frac{l_u}{R_u} + \frac{c_u l_u}{f_u^C} \leq T_u, u \in \mathcal{U}, \tag{19}$$

$$\text{C2}: \frac{c_u(d_u - l_u)}{f_u^l} \leq T_u, u \in \mathcal{U}, \tag{20}$$

$$\text{C3}: l_u \leq d_u, \forall u \in \mathcal{U}, \tag{21}$$

$$\text{C4}: 0 \leq w_u \leq 1, \forall u \in \mathcal{U}, \tag{22}$$

$$\text{C5}: \mathsf{CVaR}_\beta(R) \leq \alpha, \tag{23}$$

$$\text{C6}: Pr[R_{urllc} \leq L] \leq \epsilon, \tag{24}$$

$$\text{C7}: \sum_{u=1}^{U} y_u^b \leq 1, \forall b \in \mathcal{B}, \tag{25}$$

$$\text{C8}: y_u^b \in \{0, 1\}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \tag{26}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

where ϕ is the weight parameter. C1 and C2 represent the execution latency constraint of the eMBB users and C3 guarantees that the offloading data size of user u has to be less than the total input data size. Constraint C4 states the weight parameter for eMBB user u that can be punctured by the traffic of URLLC users, while constraint C5 and C6 show the reliability constraints of eMBB users and URLLC users, respectively. Constraint C7 ensures that one resource block can be allocated to only one eMBB user according to the OFDMA concept. However, each eMBB user can be assigned to more than one resource block. Finally, C8 represents the binary variable for the resource block allocation. The structure of the aforementioned problem is non-convex because of the non-convexity nature of the energy function E_u^{Off} in objective function (18) and the decision variables such as $\boldsymbol{y}$, $\boldsymbol{l}$, and $\boldsymbol{w}$ which are coupling to each other in both objective function and constraints. Moreover, it is a MINLP (mixed integer non-linear problem) because the problem is mixed with the continuous variables ($\boldsymbol{l}$, $\boldsymbol{w}$) and the binary variable $\boldsymbol{y}$. Therefore, solving the problem mentioned above will take exponential time complexity.

IV. PROPOSED BLOCK COORDINATE DESCENT BASED SOLUTION APPROACH

In order to address our proposed problem, firstly, we relax the resource block allocation variable in the constraint C8 into a continuous form, i.e., $0 \leq y_u^b \leq 1$. Therefore, the problem in (18) can be rewritten as follows:

$$\min_{\boldsymbol{y},\boldsymbol{l},\boldsymbol{w}} \left(\sum_{u=1}^{U} E_u^{\text{Off}} + \sum_{u=1}^{U} E_u^{L} \right) - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b} \quad (27)$$

$$\text{s.t. (C1)-(C7)}, \quad (28)$$

$$\text{C8} : y_u^b \in [0,1], \forall b \in \mathcal{B}, \forall u \in \mathcal{U}. \quad (29)$$

However, the problem is still non-convex because the decision variables are coupling in both the objective function and the constraints. Even though the problem is non-convex, as an example, the subproblem of optimizing variable $\boldsymbol{y}$ while fixing the remaining variables such as $\boldsymbol{l}$, and $\boldsymbol{w}$ is convex. As a conclusion, the proposed problem is in the form of a multi-convex problem. Therefore, to solve the proposed multi-convex problem, we decompose the proposed problem into multiple subproblems such as 1) optimal task offloading problem, 2) resource blocks allocation for eMBB users, and 3) traffic of URLLC users scheduling problem. Then, BCD algorithm is adopted to solve the subproblems alternately.

For any given $\boldsymbol{y}$, and $\boldsymbol{w}$, the optimal task offloading problem can be presented as follows:

$$\text{(P1)} : \quad \min_{\boldsymbol{l}} \sum_{u=1}^{U} E_u^{\text{Off}} + \sum_{u=1}^{U} E_u^{L} \quad (30)$$

$$\text{s.t. (C1)-(C3)}, \quad (31)$$

where (P1) is the convex problem.

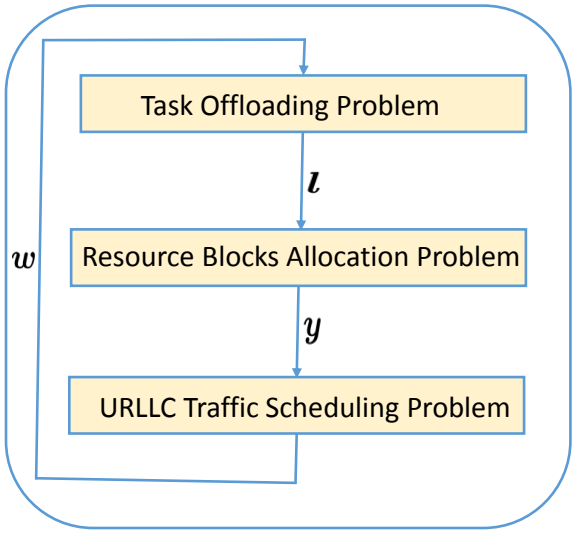


FIGURE 3: Optimization Framework.

Proof: We first set $A(\boldsymbol{l}) = \sum_{u=1}^{U} E_u^{\text{Off}} + \sum_{u=1}^{U} E_u^{L}$. Then, the first-order derivative of $A(\boldsymbol{l})$ with respect to the l_u as follows:

$$\frac{dA(\boldsymbol{l})}{dl_u} = \sum_{b=1}^{B} y_u^b \frac{P_u^b}{R_u^b} - k(f_u^l)^2 c_u, \quad \forall u \in \mathcal{U}. \quad (32)$$

Moreover, the second-order derivative is:

$$\frac{d^2 A(\boldsymbol{l})}{dl_u^2} = 0. \quad (33)$$

From (33), it is sure that the second order derivative of the objective function $A(\boldsymbol{l})$ is positive semi-definite. Therefore, we can conclude that the the objective function in (30) is convex. Moreover, the constraints (C1)-(C3) are linear constraints. Therefore, the task offloading problem, (P1), is a convex problem.

■

Secondly, for any given $\boldsymbol{l}$, and $\boldsymbol{w}$, the resource blocks allocation problem can be described as follows:

$$\text{(P2)} : \quad \min_{\boldsymbol{y}} \sum_{u=1}^{U} E_u^{\text{Off}} - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b} \quad (34)$$

$$\text{s.t. (C1), (C7), and (C8)}, \quad (35)$$

where (P2) is the convex problem.

Proof: Let $G(\boldsymbol{y}) = \sum_{u=1}^{U} E_u^{\text{Off}} - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b}$. The first-order derivative of $G(\boldsymbol{y})$ with respect to y_u^b as follows:

$$\frac{dG(\boldsymbol{y})}{dy_u^b} = P_u^b \frac{l_u}{R_u^b} - \phi \sum_{u=1}^{U} R_{u,b}, \quad \forall u \in \mathcal{U}, \forall b \in \mathcal{B}. \quad (36)$$

Then, the second-order derivative is given by:

$$\frac{d^2 G(\boldsymbol{y})}{d(y_u^b)^2} = 0, \quad (37)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

where (37) demonstrates that $G(\boldsymbol{y})$ is positive semidefinite. Therefore, we can conclude that the objective function of the resource blocks allocation problem in (34) is convex. Let $g(y_u^b) = \frac{l_u}{R_u} + \frac{c_u l_u}{f_u^C} - T_u$, then, we can rewrite the constraint C2 as follows:

$$g(y_u^b) \leq 0, u \in \mathcal{U}. \tag{38}$$

The first order derivative of $g(y_u^b)$ with respect to y_u^b is as follows:

$$\frac{dg(y_u^b)}{dy_u^b} = \frac{-l_u}{(y_u^b)^2 \omega \left(1 - w_u \times \frac{L}{L_{\max}}\right) \log_2 \left(1 + \frac{P_u^b g_u^b}{N_0}\right)}. \tag{39}$$

Then,

$$\frac{d^2 g(y_u^b)}{d(y_u^b)^2} = \frac{2l_u}{(y_u^b)^3 \omega \left(1 - w_u \times \frac{L}{L_{\max}}\right) \log_2 \left(1 + \frac{P_u^b g_u^b}{N_0}\right)}. \tag{40}$$

From (40), we can see that $\frac{d^2 g(y_u^b)}{d(y_u^b)^2} > 0$. Therefore, we can conclude that the constraint, C2, is convex [34]. Furthermore, the constraints C7 and C8 are affine. Therefore, the conclusion is that the resource blocks allocation problem, (P2), is a convex problem. ■

Then, the traffic of URLLC users scheduling problem at given $\boldsymbol{l}$, and $\boldsymbol{y}$ can be formulated as:

$$(\text{P3}): \quad \min_{\boldsymbol{w}} \sum_{u=1}^{U} E_u^{\text{Off}} - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b} \tag{41}$$

$$\text{s.t. (C4), (C5), and (C6),} \tag{42}$$

where the traffic of URLLC users scheduling problem is a convex problem.

Proof: Let $H(\boldsymbol{w})$ be $\sum_{u=1}^{U} E_u^{\text{Off}} - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b}$. The first-order derivative of $H(\boldsymbol{w})$ with respect to the w_u is given by:

$$\frac{dH(\boldsymbol{w})}{dw_u} = \frac{L}{L_{\max}} \sum_{b=1}^{B} \frac{y_u^b P_u^b l_u}{\omega (1 - w_u \times \frac{L}{L_{\max}})^2 \log_2 (1 + \frac{P_u^b g_u^b}{N_0})} + \frac{\phi L}{L_{\max}} \sum_{b=1}^{B} y_u^b \omega \log_2 (1 + \frac{P_u^b g_u^b}{N_0}), \forall u \in \mathcal{U}. \tag{43}$$

Then, the second-order derivative is as follows:

$$\frac{d^2 H(\boldsymbol{w})}{d(w_u)^2} = \frac{2L^2}{L_{\max}^2} \sum_{b=1}^{B} \frac{y_u^b P_u^b l_u}{\omega (1 - w_u \times \frac{L}{L_{\max}})^3 \log_2 (1 + \frac{P_u^b g_u^b}{N_0})}. \tag{44}$$

From (44), we notice that $\frac{d^2 H(\boldsymbol{w})}{d(w_u)^2} > 0$. Therefore, the objective function of the URLLC users' traffic scheduling problem is a convex function. In addition, the constraint C4 is affine, and C5 and C6 are convex. Hence, the URLLC users' traffic scheduling problem is a convex problem. ■

Algorithm 1 Iterative Algorithm for the relaxed problem

1: **Initialization:** Set $k = 0$, $\epsilon_1, \epsilon_2, \epsilon_3 > 0$, and initial solutions $(\boldsymbol{l}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{w}^{(0)})$;
2: **repeat**
3: Compute $\boldsymbol{l}^{(k+1)}$ from (P1) at given $\boldsymbol{y}^k$, and $\boldsymbol{w}^k$;
4: Compute $\boldsymbol{y}^{(k+1)}$ from (P2) at given $\boldsymbol{l}^{(k+1)}$, and $\boldsymbol{w}^k$;
5: Compute $\boldsymbol{w}^{(k+1)}$ from (P3) at given $\boldsymbol{l}^{(k+1)}$, and $\boldsymbol{y}^{(k+1)}$;
6: $k = k + 1$;
7: **until** $\| \boldsymbol{l}^{(k+1)} - \boldsymbol{l}^{(k)} \| \leq \epsilon_1$, $\| \boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)} \| \leq \epsilon_2$, and $\| \boldsymbol{w}^{(k+1)} - \boldsymbol{w}^{(k)} \| \leq \epsilon_3$;
8: Then, set $\left(\boldsymbol{l}^{(k+1)}, \boldsymbol{y}^{(k+1)}, \boldsymbol{w}^{(k+1)}\right)$ as the desired solution.

We can see that all of the above subproblems are convex problems. Therefore, we can use ECOS solver in the CVXPY to solve each subproblem.

A. COMPLEXITY OF PROPOSED SOLUTION

Although the objective function and feasible set are non-convex in multiple blocks of variables, they are convex in each block of variables. It means that when we fix two among three blocks of variables, the problem becomes convex in the remaining block of variables, and it is called a multi-convex problem [35]. To solve this kind of multi-convex problem, we apply the block coordinate descent (BCD) algorithm and it guarantees to converge to the stationary point (i.e., sub-optimal solution). According to [36], with the flexible block update rule, the BCD algorithm has a sub-linear convergence rate, $\mathcal{O}(1/k)$, where k is the index of iteration.

Likewise, even though our proposed problem is non-convex, as an example, the subproblem of optimizing the block of variable $\boldsymbol{y}$ while fixing the remaining blocks of variables such as $\boldsymbol{l}$, and $\boldsymbol{w}$ is convex. As a conclusion, the proposed problem is in the form of a multi-convex problem. The computation complexity of our proposed block coordinate descent based solution approach, Algorithm 1, in a single iteration is $\mathcal{O}(|\mathcal{U}| \times |\mathcal{B}|)$ which depends on the dimensions of the blocks of variables in each subproblem. Therefore, the total complexity (i.e., execution time) of the proposed algorithm is $K \times \mathcal{O}(|\mathcal{U}| \times |\mathcal{B}|)$ where K is the total number of iterations that the proposed algorithm takes to converge to the suboptimal solution.

V. SIMULATION RESULTS

In this section, we evaluate the performance of our proposed iterative algorithm-based communication and computation resource slicing for eMBB and URLLC users in 5G networks.

A. SIMULATION SETUP

To figure out the performance of our proposed scheme, we consider a multiuser MEC system with a single BS where the MEC server having a maximum computation capacity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

TABLE 2: Summary of Simulation Parameters.

Simulation Parameters	Values
Number of BS	1
Coverage radius of BS	1000 m
Number of eMBB users	[5,30]
Carrier frequency	2 GHz
Frame structure	FDD
Total system bandwidth to allocate eMBB users	10 MHz
Number of resource blocks	[30,50]
System bandwidth of each resource block	150 KHz
Thermal noise density	-174 dBm/Hz
Fading model	Rayleigh fading
Input data size of eMBB user	[0.1, 0.4] MB
Required CPU cycles to execute one bit of data	[10, 25] cycles
Maximum tolerable latency of each eMBB user	[30, 100] s
Maximum computation capacity of each eMBB user	[0.1, 0.72] MHz
Maximum computation capacity of MEC server	1 GHz
Convergence threshold $(\epsilon_1, \epsilon_2, \epsilon_3)$	10^{-4}

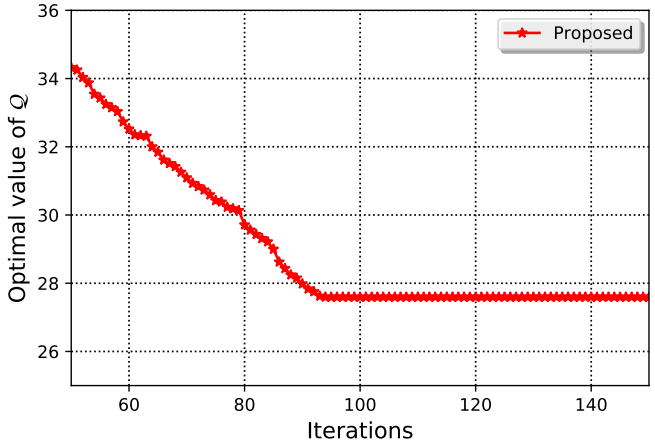


FIGURE 4: Convergence rate of proposed algorithm.

of 1GHz is deployed. The maximum system bandwidth of the BS is 10MHz. The eMBB users are randomly scattered within the radius of the cell $r = 1000$ m, and every eMBB user has computation tasks to be executed. The input data size of tasks of eMBB users and the required CPU cycles to execute one bit of input data are randomly generated between [0.1, 0.4] MB and [10, 25] cycles, respectively. Then, the maximum tolerable latency to execute a computation task of each user is between [30, 100] s. We set the maximum computation capacity of eMBB users between [0.1, 0.72] MHz and the maximum transmit power of each eMBB user as 100 mW. In this work, we consider a small scale fading model, and the back noise of the system is -174 dBm/Hz. The details of the parameters used in our simulation are presented in Table II.

B. NUMERICAL RESULTS

In this section, we focus primarily on the proposed algorithm's performance gain. In addition, we compare the efficiency of our proposed iterative algorithm with the following two baseline schemes.

- Equal resource sharing: The communication resource (i.e., resource blocks) and the computation capacity (i.e., CPU capacity (cycles/s)) of the MEC server are equally allocated to all eMBB users.
- All local computing: In this scheme, all eMBB users execute their computation tasks locally.
- All offloading: In this scheme, all eMBB users offload all of their computation tasks to the MEC server.

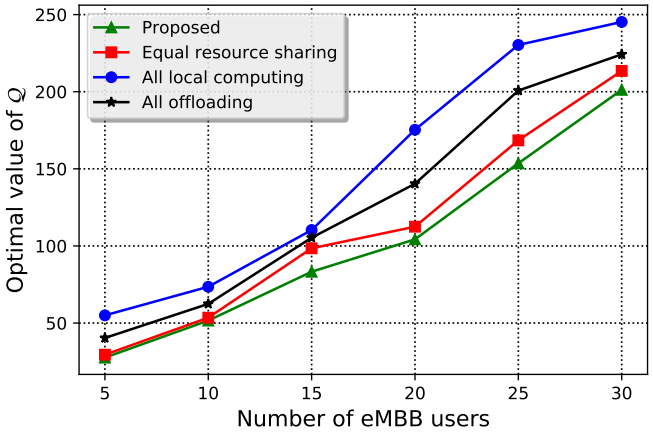


FIGURE 5: Optimal value of $\mathcal{Q}$ under different number of users.

In Fig. 4, we demonstrate the converge rate of the proposed algorithm with 5 eMBB users where $\mathcal{Q} = \sum_{u=1}^{U} E_u^{\text{Off}} + \sum_{u=1}^{U} E_u^L - \phi \sum_{u=1}^{U} \sum_{b=1}^{B} y_u^b R_{u,b}$. From there, we observe that our proposed algorithm takes 95 iterations to converge to the optimal solution. In Fig. 5, the optimal value of $\mathcal{Q}$ for a different number of eMBB users is described. Moreover, we compare the performance of our proposed algorithm with other schemes such as equal resource sharing, all local computing, and all offloading. Under equal resource sharing scheme, the BS allocates its physical resource blocks and the computation capacity of the MEC server to all eMBB users equally, and all eMBB users execute their computation tasks locally and offload their computation tasks to the MEC server under all local computing and all offloading schemes. From Fig. 5, we observe that our proposed scheme outperforms the other three schemes. The performance gap between our proposed scheme and other schemes is higher when the number of users increases. Therefore, our proposed scheme is more efficient when there is a large number of users in the system.

Fig. 6 compares the total energy consumption under different number of users through different algorithms: equal resource sharing, all local computing, all offloading and our proposed algorithm. From Fig. 6, we observe that the energy consumption is minimum under our proposed scheme. The reason is that the equal allocation of resource blocks and computation capacity of the MEC server cannot be effective for all eMBB users. This is because all eMBB users have different achievable channel gains and different sizes of offloaded data. Therefore, more resource blocks are needed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

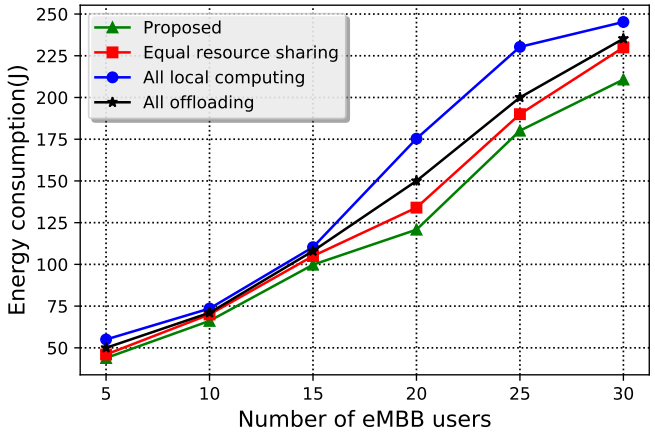


FIGURE 6: Energy consumption under different number of users.

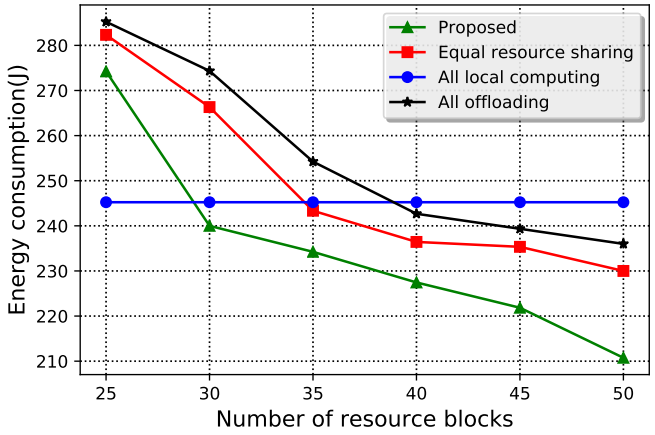


FIGURE 7: Energy consumption under different number of resource blocks.

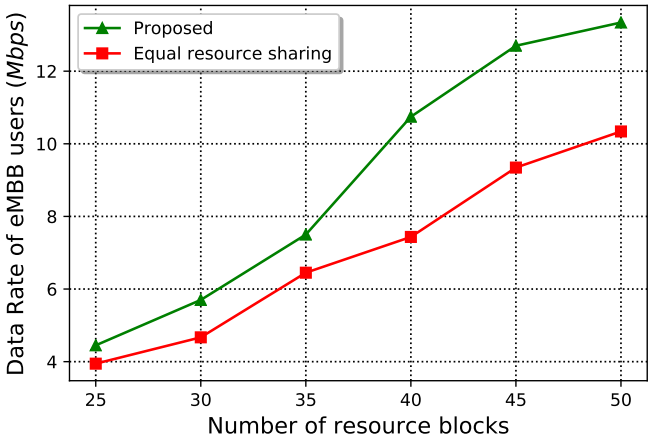


FIGURE 8: Data rate of eMBB users under different number of resource blocks.

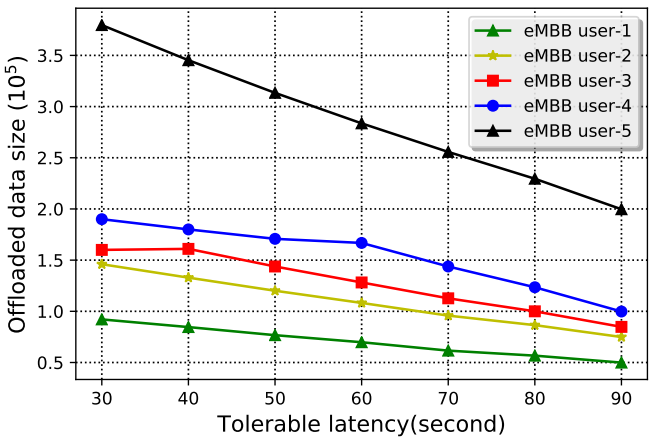


FIGURE 9: Offloaded data size under different tolerable latencies.

to be allocated to lower channel gain users and users with the higher offloaded data size need to get more fraction of the CPU capacity of the MEC server. Furthermore, there will not be enough resource blocks to allocate to all eMBB users when users offload all of their computation tasks to the MEC server. Therefore, all offloading scheme is not efficient when compared with our proposed algorithm. In addition, all users' battery life is limited, and so it is not effective for the users to perform all of their computation tasks locally.

Furthermore, Fig. 7 shows the energy consumption of the eMBB users under different number of resource blocks. From Fig. 7, we may note that, under all local computing scheme, energy consumption is the same even though the number of resource blocks increases. This is because all users do not offload their computation tasks to the MEC server and execute their computation tasks locally under all local computing scheme, therefore increasing the number of resource blocks does not impact the users' energy usage. Moreover, when the number of resource blocks is small, the users' energy consumption under proposed algorithm is high. However, when the number of resource blocks is large, our proposed algorithm does perform better. This is because a user's energy consumption decreases according to (9) when the user's data rate increases and the user's data rate increases when the number of resource blocks allocated to it increases per (5). In conclusion, it is clear from Fig. 7 that our proposed algorithm outperforms equal sharing and all offloading schemes. Moreover, our proposed algorithm performs far better than all local computing scheme when the number of resource blocks is sufficiently large. In addition, Fig. 8 provides the total achievable data rate of the eMBB users under different number of resource blocks. From there, we can note that the data rate increases with an increase in the number of resource blocks. Finally, it is clear in Fig. 8 that the attainable data rate under our proposed algorithm is higher than that under equal sharing scheme.

Fig. 9 demonstrates the offloaded data size of the task from each eMBB user to the MEC server. From Fig. 9, we observe

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

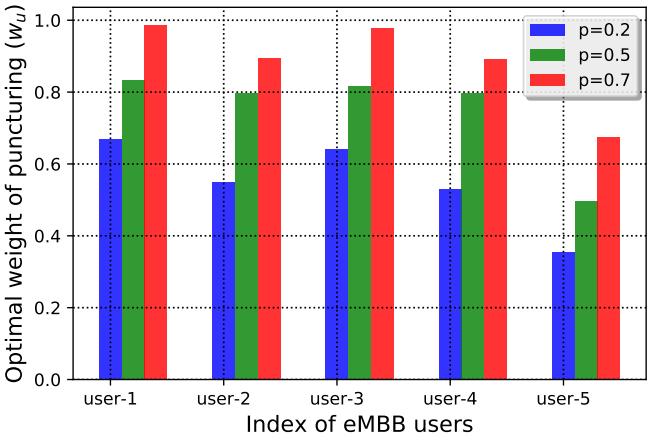


FIGURE 10: Optimal weight of puncturing resources of eMBB users.

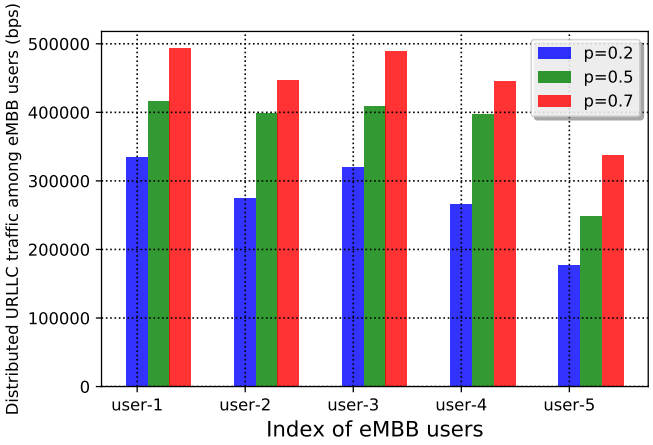


FIGURE 11: Data rate of URLLC users on the puncturing resource of eMBB users.

that users offload more data to the server when the tolerable latency (i.e., computation deadline) of the task is low. It is because each user has limited computation capacity when compared with the MEC server, and it is difficult for each user to complete their task execution within the deadline. When the tolerable latency is high enough, each eMBB user will offload less data to the server and will do more local computing.

We show the optimal weight of puncturing resources of each eMBB users under different traffic of URLLC users (i.e., different p) in Fig. 10. We observe that the resource of the eMBB users has been punctured more when the incoming traffic of URLLC users is high (i.e., higher p). As an example, the allocated resource of the eMBB user-1 will be punctured more in $p = 0.7$ when compared with $p = 0.2$ and $p = 0.5$. Moreover, we see that resources from the eMBB user-1 must be punctured more than from eMBB user-5. The reason is that the achievable channel gain of user-1 is higher than that of user-5 or the QoS requirement of user-5 is higher than that of user-1. Then, Fig. 11 presents the achievable data rate of the URLLC users on the puncturing resource of each eMBB user. We see that the achievable data rate of the URLLC users on the puncturing resource of user-1 is higher than others because more resource from the user-1 is punctured for the traffic of URLLC users. Moreover, we can observe that the achievable data rate of the URLLC users is the highest at $p = 0.7$ (i.e., highest traffic of URLLC users). Finally, we demonstrate the achievable data rate of eMBB users in Fig. 12. From Fig. 12, we observe that the achievable data rate of user-5 is the highest compared with other eMBB users. The reason is that the allocated resource of user-1 is less punctured when compared with the resources of other users. Moreover, it can be seen that the achievable instantaneous data rate of all eMBB users is lowest when the incoming traffic of URLLC users is the highest (i.e., highest p). The fact is that more resources of the eMBB users will be punctured when the incoming traffic of URLLC users is high.

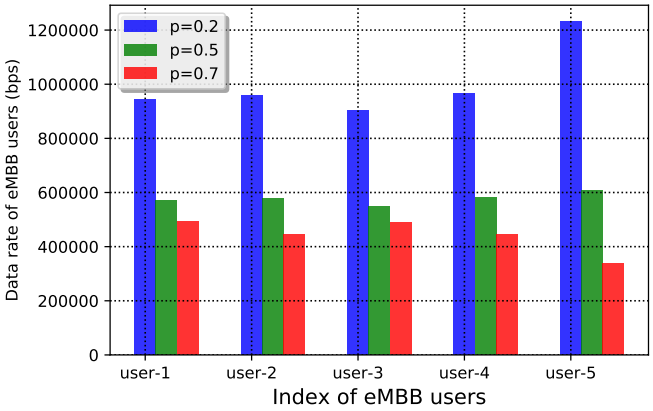


FIGURE 12: Achieved data rate of eMBB users.

VI. CONCLUSIONS

In this work, we have formulated an energy-efficient joint communication and computation resource allocation problem for eMBB, and URLLC users in 5G networks. Then, we have decomposed the formulated problem into multiple sub-problems and solved them alternately until convergence. The superior effectiveness in the performance of our proposed algorithm over other existing schemes has been proven with simulation results. In the future, we will consider multiple base stations scenario and take into account the power control of the users.

REFERENCES

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

[2] L. Liu, Y. Du, Q. Fan, and W. Zhang, "A survey on computation offloading in the mobile cloud computing environment," International Journal of Computer Applications in Technology, vol. 59, no. 2, pp. 106–113, Mar. 2019.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on

mobile edge computing: The communication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.

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

[5] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1397–1411, Dec. 2016.

[6] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2633–2645, Dec. 2017.

[7] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[8] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for internet of things," IEEE Transactions on Cloud Computing, pp.1-1, Feb. 2019.

[9] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "embb-urllc resource slicing: A risk-sensitive approach," IEEE Communications Letters, vol. 23, no. 4, pp. 740–743, Feb. 2019.

[10] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of urllc and embb traffic in 5G wireless networks," in Proc. of IEEE Conference on Computer Communications (INFOCOM), pp. 1970-1978, Honolulu, HI, USA, Oct. 2018.

[11] A. A. Esswie and K. I. Pedersen, "Opportunistic spatial preemptive scheduling for urllc and embb coexistence in multi-user 5G networks," IEEE Access, vol. 6, pp. 38 451–38 463, July 2018.

[12] M. Alsenwi, S. R. Pandey, Y. K. Tun, K. T. Kim, and C. S. Hong, "A chance constrained based formulation for dynamic multiplexing of embb-urllc traffics in 5G new radio," in Proc. of IEEE International Conference on Information Networking (ICOIN), pp. 108–113, Kuala Lumpur, Malaysia, May 2019.

[13] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," IEEE Transactions on Wireless Communications, vol. 17, no. 3, pp. 1784–1797, 2017.

[14] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5g," IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 6398–6409, 2018.

[15] K. Li, M. Tao, and Z. Chen, "Exploiting computation replication for mobile edge computing: A fundamental computation-communication tradeoff study," IEEE Transactions on Wireless Communications, 2020.

[16] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," IEEE Wireless Communications Letters, vol. 6, no. 6, pp. 774–777, 2017.

[17] L. Chen, S. Zhou, and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," in 2017 IEEE International Conference on Communications (ICC). IEEE, 2017, pp. 1–6.

[18] T. Zhang, "Data offloading in mobile edge computing: A coalition and pricing based approach," IEEE Access, vol. 6, pp. 2760–2767, 2017.

[19] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2082–2091, 2017.

[20] Z. Lan, W. Xia, W. Cui, F. Yan, F. Shen, X. Zuo, and L. Shen, "A hierarchical game for joint wireless and cloud resource allocation in mobile edge computing system," in 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2018, pp. 1–7.

[21] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, "A hierarchical game framework for resource management in fog computing," IEEE Communications Magazine, vol. 55, no. 8, pp. 52–57, 2017.

[22] Q.-V. Pham, T. Leanh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach," IEEE Access, vol. 6, pp. 75 868–75 885, 2018.

[23] R. Kassab, O. Simeone, and P. Popovski, "Coexistence of urllc and embb services in the c-ran uplink: an information-theoretic study," in 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1–6.

[24] R. Abreu, T. Jacobsen, G. Berardinelli, K. Pedersen, N. H. Mahmood, I. Z. Kovács, and P. Mogensen, "On the multiplexing of broadband traffic and grant-free ultra-reliable communication in uplink," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring). IEEE, 2019, pp. 1–6.

[25] P. Yang, X. Xi, Y. Fu, T. Q. Quek, X. Cao, and D. Wu, "Multicast embb and bursty urllc service multiplexing in a comp-enabled ran," arXiv preprint arXiv:2002.09194, 2020.

[26] W.-R. Wu, P.-Y. Lin, and Y.-H. Lee, "An indicator-free embb and urllc multiplexed scheme for 5g downlink system," in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall). IEEE, 2019, pp. 1–5.

[27] Y. Li, C. Hu, J. Wang, and M. Xu, "Optimization of urllc and embb multiplexing via deep reinforcement learning," in 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops). IEEE, 2019, pp. 245–250.

[28] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for embb and urllc coexistence in 5g and beyond: A deep reinforcement learning based approach," arXiv preprint arXiv:2003.07651, 2020.

[29] H. Ding, Y. Zhang, L. Xia, and Q. Wang, "Use cases and practical system design for urllc from operation perspective," in 2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2019, pp. 1–6.

[30] M. Elsayed and M. Erol-Kantarci, "Ai-enabled radio resource allocation in 5g for urllc and embb users," in 2019 IEEE 2nd 5G World Forum (5GWF). IEEE, pp. 590–595.

[31] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," IEEE Access, vol. 6, pp. 55 765–55 779, Sep. 2018.

[32] Y. K. Tun, S. R. Pandey, M. Alsenwi, C. W. Zaw, and C. S. Hong, "Weighted proportional allocation based power allocation in wireless network virtualization for future wireless networks," in Proc. of IEEE International Conference on Information Networking (ICOIN), pp. 284–289, Luala Lumpur, Malaysia, May 2019.

[33] R. T. Rockafellar, S. Uryasev et al., "Optimization of conditional value-at-risk," Journal of risk, vol. 2, pp. 21–42, April 2000.

[34] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.

[35] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," SIAM Journal on imaging sciences, vol. 6, no. 3, pp. 1758–1789, 2013.

[36] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, "Iteration complexity analysis of block coordinate descent methods," Mathematical Programming, vol. 163, no. 1-2, pp. 85–114, 2017.

YAN KYAW TUN received the B.E. degree in marine electrical systems and electronics engineering from Myanmar Maritime University, Thanlyin, Myanmar, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University, South Korea, for which he received a scholarship, in 2015. His research interests include network economics, game theory, network optimization, wireless communication, wireless network virtualization, mobile edge computing, and wireless resource slicing for 5G.

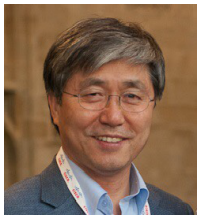
DO HYEON KIM received his B.S. degree in Communication Engineering from Jeju National University, in 2014 and received M.S. degree from Kyung Hee University in 2017. He is currently working toward his Ph.D. degree at the Department of Computer Science and Engineering, Kyung Hee University. His research interests include Multi-access Edge Computing, Wireless Network Virtualization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3011167, IEEE Access

MADYAN ALSENWI received the B.E. and M.Sc. degrees in electronics and communications engineering from Cairo University, Egypt, in 2011 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University, South Korea. He was a Research Assistant under several research projects funded by the Egyptian Government. His research interests include wireless communications and networking, resource slicing in 5G wireless networks, ultra reliable low latency communications (URLLC), UAV-assisted wireless networks, and machine learning.

NGUYEN H. TRAN (S'10-M'11-SM'18) received the BS degree from Hochiminh City University of Technology and Ph.D degree from Kyung Hee University, in electrical and computer engineering, in 2005 and 2011, respectively. He was an Assistant Professor with Department of Computer Science and Engineering, Kyung Hee University from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, where he is currently a Senior Lecturer. His research interest is distributed computing and learning over networks. He received the best KHU thesis award in engineering in 2011 and several best paper awards, including IEEE ICC 2016, APNOMS 2016, and IEEE ICCS 2016. He receives the Korea NRF Funding for Basic Science and Research from 2016 to 2023. He has been the Editor of IEEE Transactions on Green Communications and Networking since 2016.

ZHU HAN (S'01-M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks".



CHOONG SEON HONG (AM'95-M'07-SM'11) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Tokyo, Japan, in 1997. In 1988, he joined KT, Gyeonggi-do, South Korea, where he was involved in broadband networks as a member of the Technical Staff. Since 1993, he has been with Keio University. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include future Internet, intelligent edge computing, network management, and network security. Dr. Hong is a member of the Association for Computing Machinery (ACM), the Institute of Electronics, Information and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Korean Institute of Information Scientists and Engineers (KIISE), the Korean Institute of Communications and Information Sciences (KICS), the Korean Information Processing Society (KIPS), and the Open Standards and ICT Association (OSIA). He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member of international conferences, such as the Network Operations and Management Symposium (NOMS), International Symposium on Integrated Network Management (IM), Asia-Pacific Network Operations and Management Symposium (APNOMS), End-to-End Monitoring Techniques and Services (E2EMON), IEEE Consumer Communications and Networking Conference (CCNC), Assurance in Distributed Systems and Networks (ADSN), International Conference on Parallel Processing (ICPP), Data Integration and Mining (DIM), World Conference on Information Security Applications (WISA), Broadband Convergence Network (BcN), Telecommunication Information Networking Architecture (TINA), International Symposium on Applications and the Internet (SAINT), and International Conference on Information Networking (ICOIN). He was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He currently serves as an Associate Editor for the International Journal of Network Management and an Associate Technical Editor of the IEEE Communications Magazine.