# Energy-Efficient Multi-Task Multi-access Computation Offloading via NOMA Transmission for IoTs

Yuan Wu *Senior Member IEEE*, Binghua Shi, Li Ping Qian *Senior Member IEEE*,
Fen Hou, Jiali Cai, Xuemin (Sherman) Shen *Fellow IEEE*

***Abstract*—Driven by the explosive growth in computation-intensive applications in future 5G networks and industries, mobile edge computing (MEC), which enables smart terminals to offload their computation-workloads to nearby edge servers (ESs) in radio access networks, has attracted increasing attentions. In this paper, we investigate the energy-efficient multi-task multi-access mobile edge computing (MEC) via non-orthogonal multiple access (NOMA). Exploiting NOMA, a smart terminal (ST) with multiple tasks can offload the respective computation-workloads of different tasks to different ESs simultaneously. To study this problem, we adopt a two-step approach. Specifically, we first consider a given task-ES assignment and formulate a joint optimization of the tasks' computation-offloading, local computation-resource allocation, and the NOMA-transmission duration, with the objective of minimizing the ST's total energy consumption for completing all tasks. Next, based on the optimal offloading solution for the given task-ES assignment, we further investigate how to properly assign different tasks to the ESs for further minimizing the ST's total energy consumption. For both the formulated problems, we propose efficient algorithms to compute the respective solutions. Numerical results are provided to validate the effectiveness of our proposed algorithms. The results also show that our proposed NOMA-enabled multi-task multi-access computation offloading can outperform conventional orthogonal multiple access (OMA) based offloading scheme, especially when the tasks have heavy computation-workload requirements and stringent delay-limits.**

## I. Introduction

In the past decades, we have witnessed an explosive growth in mobile Internet services along with the growing deployment of 4G/5G cellular networks, leading to a portfolio of resource-hungry applications, e.g., unmanned vehicles, online artificial intelligence, virtual/augmented reality, and industrial Internet of Things (IoTs). However, conventional smart IoT terminals usually suffer from limited computation-resources, which result in degraded quality of experience (e.g., excessive delay) when running these resource-hungry applications locally. Mobile edge computing (MEC), which enables the resource-limited smart terminals (STs) to offload their computation-tasks to the edge-servers (ESs) equipped with sufficient computation-resources, has provided an effective approach to address this issue. Thanks to its potentials in reducing computation-delay and improving resource-utilization efficiency, MEC has attracted lots of attentions from both academia and industries [1]–[6]. To further improve the efficiency of MEC, the paradigm of multi-access MEC has been envisioned. In multi-access MEC, a ST can offload its computation-tasks to several ESs simultaneously, yielding a more efficient utilization of ESs' resources [7]–[9].

However, the success of MEC and multi-access MEC necessitates a joint optimization of computation-workload offloading, computation-resource allocation, as well as the transmission-resource allocation. Many studies have been devoted to investigating the joint resource management scheme for MEC and multi-access MEC. In [10], [11], joint offloading decision and channel allocations have been studied. In [12], Chen *et. al.* further considered a multi-task scenario and aimed at jointly optimizing the offloading decision and allocations of computation and communication resources. In [13], taking into account the time-varying network conditions, a learning-based dynamic computation-offloading policy for multi-cell MEC systems has been proposed. In [14], an online computation offloading scheme has been proposed for wireless powered MEC. In [15], Dinh *et. al.* adopted the game-theoretic approach to study the multi-user multi-edge-node offloading problem. For reducing energy consumption of MEC, green-oriented MEC via joint optimization of computation offloading and energy management has been studied in [17]–[19]. The authors of [20] investigated joint management scheme of computation offloading and resource allocation from the perspective of network economics. In [16], Zhang *et. al.* studied the computation resource management problem in mobile edge-cloud computing networks for sharing the resources between edge systems and cloud networks. Exploiting MEC, in [21], Xu *et. al.* proposed a blockchain-based non-repudiation

Y. Wu is with State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China, and also with Department of Computer and Information Science, University of Macau (email: ywucisum@gmail.com).

L. Qian, B. Shi, and J. Cai are with College of Information Engineering, Zhejiang University of Technology, Hangzhou, China (emails:lpqian@zjut.edu.cn, bhshi_zjut@163.com, jlcai_zjut@163.com). L. Qian is also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, 210096, China. L. Qian is the corresponding author.

F. Hou is with State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China, and also with Department of Electrical and Computer Engineering, University of Macau (email:fenhou@um.edu.mo).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail:xshen@bbcr.uwaterloo.ca).

This work was supported in part by the National Natural Science Foundation of China under Grant 61572440, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR16F010003, in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No.2019D11), in part by the Research Grant of University of Macau under Grant SRG2019-00168-IOTSC, and in part by the Natural Sciences and Engineering Research Council, Canada.

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics

network computing scheme for industrial IoT. In [22], to address the limited capability of lightweight IoT devices, a block-streaming application execution scheme based on edge computing has been proposed. To exploit multi-access MEC, Guo *et. al.* considered a scenario of multi-user ultra-dense MEC servers and proposed a greedy offloading scheme in [9]. In [23], an integrated scheme for multi-access edge computing and fiber-wireless access networks has been proposed. User-association to different resources has been studied in [24] for multi-access MEC.

Recently, non-orthogonal multiple access (NOMA) has been considered as one of the enabling technologies for achieving ultra high-throughput and accommodating massive connectivity in radio access networks (RANs) [25]–[27]. Many studies have been devoted to investigating the resource management for NOMA to exploit its benefits, e.g., for enhancing throughput [28], [29] and energy-efficiency [30]. In particular, NOMA has been envisioned as a promising scheme for enabling the multi-access MEC. Exploiting NOMA, a ST can simultaneously send its offloaded workloads to multiple edge-servers over the same frequency channel, which thus may help reduce the offloading delay and energy consumption. For instance, to reduce the energy consumption in the context of edge computing, in [31], Kiani *et. al.* proposed a NOMA-based optimization framework that jointly optimizes the user clustering, computing and communication resource allocation, and transmit powers for minimizing the energy consumption of MEC users. To reduce the computation-delay, in [32], Ding *et. al.* investigated the minimization of the offloading delay for NOMA assisted MEC and established the criteria for choosing among different offloading-transmission modes for a two-user scenario. In [33], a multi-user NOMA-enabled MEC scheme has been proposed for minimizing the overall delay in MEC (including the tasks' computation-delay as well as the uploading and downloading transmission-delay).

In many industrial applications, a smart agent may have a group of tasks to be processed in parallel, with different tasks having different computation-workload requirements and different delay-limits. For instance, in the surveillance system of an unmanned factory, a smart camera may need to execute the delay-tolerant task of video-data compression and the delay-sensitive task of realtime video analytics. Also, in the context of automotive driving, an automotive vehicle may execute realtime computation for target identification as well as mobile data services which are delay-tolerant. Therefore, taking into account that i) different tasks may have different delay-limits and different workload-requirements, and ii) different ESs may have different computation-rates to process the offloaded workloads, it is a critical issue about how to properly exploit the tasks' different delay-limits and the ESs' different computation-rates for optimizing the performance of the multi-task computation offloading (e.g., for minimizing the ST's total energy consumption for completing all tasks). In particular, this issue becomes even more challenging when we exploit NOMA in the offloading. Although NOMA enables the simultaneous transmissions of different tasks' offloaded workloads to the respective ESs, the resulting co-channel interference among these offloading-transmissions will strongly couple the offloading-decisions (e.g., the amount of the offloaded workloads) of different tasks. These, however, have not been investigated in the existing studies yet. Driven by these motivations, we investigate the energy efficient multi-task computation offloading via NOMA. Our detailed contributions in this work are summarized as follows.

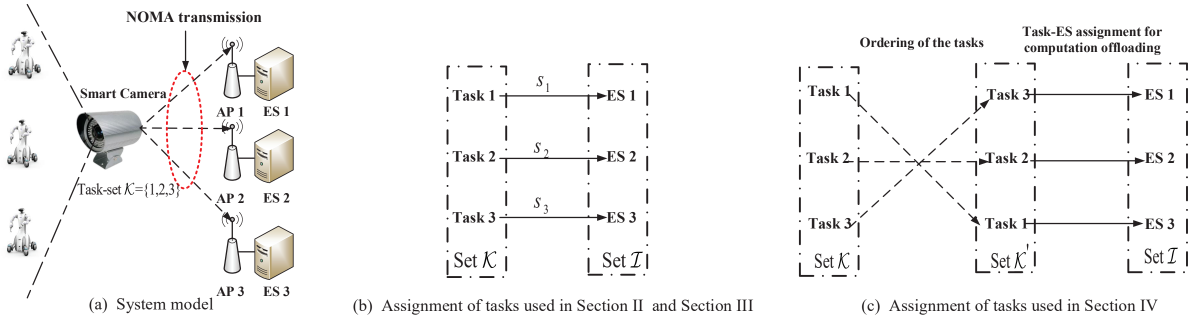
- *(Problem formulation):* We study the NOMA enabled multi-task multi-access MEC with the objective of minimizing the ST's total energy for completing all tasks, while subject to each task's delay-limit. To investigate the problem, we adopt a two-step approach. Specifically, in the first step, we consider a given task-ES assignment and formulate a joint optimization of the tasks' computation-offloading, the ST's local computation-rate allocation, and the NOMA-transmission time allocation. Based on the optimal offloading solution in the first step we further investigate the optimal assignment of the tasks to different ESs (i.e., the task-ES assignment), with the objective of further minimizing the ST's total energy consumption.
- *(Solution methodology for the first-step problem):* Despite the non-convexity of the formulated joint optimization problem in the first step, we exploit its layered structure and propose an efficient layer-algorithm to compute the optimal offloading solution. To validate our proposed algorithm, we compare the solution of our proposed algorithm with that of LINGO [36] (a commercial optimization package), in terms of the accuracy and efficiency.
- *(Solution methodology for the second-step problem):* For the optimal task-ES assignment in the second step, we treat it as an equivalent optimal ordering problem and propose an efficient index-swapping algorithm to determine the ordering of the tasks, which correspondingly gives the task-ES assignment. We finally validate the advantages of our proposed NOMA-enabled multi-task multi-access MEC scheme by providing the performance comparison with the conventional orthogonal multiple access (OMA) based MEC scheme.

## II. System Model and Problem Formulation

Figure 1(a) shows an illustrative model considered in this paper. One ST (e.g., a smart camera) is running a group of tasks (e.g., the delay-tolerable video-compression application and realtime target identification) in parallel. We denote the group tasks as $\mathcal{K} = \{1, 2, ..., K\}$, with task $k$ having a required computation-workload $S_k^{\text{tot}}$ to be completed and a delay-limit $T_k^{\max}$. Notice that different tasks can have different required computation-workloads and delay-limits. Meanwhile, there exists a group of ESs $\mathcal{I} = \{1, 2, ..., I\}$[1], with each ES co-located with a wireless access point (AP) and providing computation-offloading to the ST. In particular, due to the feature of NOMA transmission, we assume that the ESs are ordered according to:

$$g_1 \geq g_2 \geq ... \geq g_I, \tag{1}$$

[1] In this work, we focus on investigating the case that the number of the ESs is equal to the number of the tasks, namely, $I = K$. Our proposed algorithms here are also applicable to other cases. For instance, for the case of $K < I$, we could add $I - K$ virtual tasks into the system model, with each virtual task having zero computation-requirement and a very large delay-limit.



(a) System model

(b) Assignment of tasks used in Section II and Section III

(c) Assignment of tasks used in Section IV

Fig. 1: (a) System model. (b) Detailed task-ES assignment in Sections II and III. (c) Ordering of the tasks in Section IV.

where $g_i$ denotes the channel power gain from the ST to ES $i$. To exploit MEC, we consider the partial offloading. Specifically, for each task $k$, its required computation-workload is divided into two parts, i.e., the computation-workload $s_k$ to be offloaded to one of the ESs, and the workload $S_k^{\text{tot}} - s_k$ to be processed locally.

*(A detailed task-ES assignment)* In Section II and Section III, we first consider a detailed task-ES assignment, namely, part of task $k$'s workload is offloaded to ES $i = k$ directly, as shown in Figure 1(b) for a concrete example of $I = K = 3$. With this detailed assignment, we investigate the joint optimization of the tasks' workload offloading, local computation-resource allocation, and NOMA-transmission time. With the proposed algorithm for finding the optimal offloading solution for this detailed task-ES assignment, in Section IV, we will further investigate how to properly assign the tasks to different ESs, which can be regarded as an optimal ordering of the tasks as shown in Figure 1(c). In this work, as the initial step for analyzing the NOMA-assisted multi-task computation-offloading, we make an assumption that each ES can only accommodate the offloaded workload from one of the ST's tasks, and each task can only select one of the ESs to offload its workload. In practice, the ESs may simultaneously process the offloaded workloads from multiple tasks, which thus further improves the efficiency of the multi-access computation offloading. This is an interesting direction for us to extend this work here.

### A. Modelling of NOMA Transmission for Offloading

Notice that in Section II and Section III, we focus on the detailed example of task $i$'s workload being offloaded to ES $i$ directly. Thus, for the sake of clear presentation, we will use the subscript $i$ to denote both the task and the ES in Section II and Section III.

Based on the principle of the power-domain NOMA and the operations of successive interference cancellation, the ST can simultaneously send the offloaded workloads $\mathbf{s} = [s_1, s_2, ..., s_I]$, which are measured in the number of bits, to the respective ESs over the same frequency channel. Then, based on (1) and Proposition 1 in [30], the ST's minimum total transmit-power can be given by:

$$P^{\text{tot}}(\mathbf{s},t) = Wn_0 \sum_{i=1}^{I} (\frac{1}{g_i} - \frac{1}{g_{i-1}}) 2^{\frac{1}{t}\frac{1}{W}\sum_{m=i}^{I} s_m} - \frac{Wn_0}{g_I}, \quad (2)$$

where $W$ denotes the channel bandwidth, and $n_0$ denotes the power spectral density of the background noise. Correspondingly, the ST's total energy consumption for NOMA-transmission is

$$E_{\text{NOMA}} = tP^{\text{tot}}(\mathbf{s},t). \quad (3)$$

Notice that to exploit the NOMA transmission, we assume that different tasks are synchronized such that parts of the workloads can be simultaneously offloaded to the ESs via the NOMA transmission. In particular, to achieve the synchronization, we need a proper scheduling scheme to group different tasks (i.e., set $\mathcal{K}$), especially when the number of the tasks is larger than the number of the ESs. Design of this scheduling scheme is an important direction for us to extend this work.

### B. Modelling of the ST's Delay and Energy Consumption

Each ES $i$ has a fixed computation-rate denoted by $\mu_{i,\text{E}}$ (with the subscript E denoting the "Edge"). Thus, the overall delay for completing task $i$ can be given by

$$d_i^{\text{ove}} = \max\{\frac{S_i^{\text{tot}} - s_i}{\mu_{i,\text{L}}}, t + \frac{s_i}{\mu_{i,\text{E}}}\}, \quad (4)$$

where $\mu_{i,\text{L}}$ denotes the ST's allocated computation-rate for processing task $i$ locally. For the sake of clear presentation in this work, we measure $\mu_{i,\text{E}}$ and $\mu_{i,\text{L}}$ by the number of bits processed per second, which is equivalent to the CPU-rate in Hz multiplied by the number of bits processed per CPU cycle. Similar to many existing studies [31], [32], we do not account for the delay for sending back the computation-result in eq. (4), since the volume of the computation-result is usually very small. According to [34], the CPU power consumption can be modelled as a cubic relationship with respect to the computation-rate. Thus, the ST's total energy consumption for completing all remaining workloads is ("LC" denotes "local computing"):

$$E_{\text{LC}} = \sum_{i=1}^{I} \frac{S_i^{\text{tot}} - s_i}{\mu_{i,\text{L}}} \rho_{\text{L}} \mu_{i,\text{L}}^3 = \sum_{i=1}^{I} \rho_{\text{L}} (S_i^{\text{tot}} - s_i) \mu_{i,\text{L}}^2, \quad (5)$$

where $\rho_{\text{L}}$ is a coefficient depending on the CPU chip architecture.

Notice that to facilitate modelling of the overall delay for completing each task (i.e., eq. (4)), we assume a simple local

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics

processing model, namely, the local computation-rate allocation $\mu_{i,\mathrm{L}}$ for processing task $i$ will be optimized only once and used throughout the execution of the task. In particular, the modelling of the delay will be much more challenging if we allow the re-adjustment of the computation-rate of task $i$ (when some other tasks are completed in advance of task $i$ and the corresponding computation-resources are released), since we need to take into account the ordering of different finishing-time of different tasks.

### *C. Problem Formulation for Energy-Efficiency Optimization*

We formulate an energy-efficiency optimization (EEO) problem to minimize the ST's total energy consumption as follows.

$$\text{(EEO):}\ \ \min E_{\text{NOMA}} + \lambda E_{\text{LC}}$$

subject to:

$$P^{\text{tot}}(\mathbf{s}, t) \leq P^{\max}, \tag{6}$$

$$\max\Big\{\frac{S_i^{\text{tot}} - s_i}{\mu_{i,\mathrm{L}}}, t + \frac{s_i}{\mu_{i,\mathrm{E}}}\Big\} \leq T_i^{\max},\ \text{ for } i = 1, 2, ..., I, \tag{7}$$

$$\sum_{i=1}^{I} \mu_{i,\mathrm{L}} \leq \mu_{\mathrm{L}}^{\max}, \tag{8}$$

$$s_i \leq \min\{S_i^{\text{tot}}, C_i^{\max}\}, \text{for } i = 1, 2, ..., I, \tag{9}$$

$$\text{variables: } \ \mathbf{s} \geq 0, \boldsymbol{\mu}_{\mathrm{L}} \geq 0, \ \text{ and } t \geq 0.$$

Parameter $\lambda$ in the objective function denotes the relative weight on the energy consumption for local computing. Vector $\boldsymbol{\mu}_{\mathrm{L}} = [\mu_{1,\mathrm{L}}, \mu_{2,\mathrm{L}}, ..., \mu_{I,\mathrm{L}}]$ denotes the allocated local computation-rates for all tasks. Constraint (6) ensures that the ST's total transmit-power cannot exceed its transmit-power capacity denoted by $P^{\max}$. Constraint (7) ensures that the overall delay for completing task $i$ cannot exceed its required delay-limit $T_i^{\max}$. Constraint (8) ensures that the sum of the ST's allocated computation-rates to all tasks cannot exceed its maximum local computation-rate denoted by $\mu_{\mathrm{L}}^{\max}$. Finally, we take into account that each ES may have a limited capability to process the offloaded workload and thus use constraint (9) to ensure that task $i$'s offloaded workload cannot exceed ES $i$'s affordable capacity denoted by $C_i^{\max}$. Problem (EEO) jointly optimizes the tasks' offloaded workloads $\mathbf{s}$, the ST's allocated local computation-rates $\boldsymbol{\mu}_{\mathrm{L}}$ for the tasks, and the NOMA-transmission duration $t$, and it is a complicated non-convex optimization problem.

## III. Proposed Algorithm for Problem (EEO)

This section presents a detailed algorithm design for solving Problem (EEO) by exploiting its layered structure. With some manipulations, (7) leads to the following two constraints:

$$\frac{S_i^{\text{tot}} - s_i}{T_i^{\max}} \ \leq \ \mu_{i,\mathrm{L}}, \ \text{ for } i = 1, 2, ..., I, \tag{10}$$

$$s_i \ \leq \ \mu_{i,\mathrm{E}}(T_i^{\max} - t), \ \text{ for } i = 1, 2, ..., I. \tag{11}$$

By using (10) to substitute $\boldsymbol{\mu}_{\mathrm{L}}$, we obtain the following equivalent form of Problem (EEO) (where letter "E" denotes "Equivalent"):

$$\text{(EEO-E):}\ \ \min t\big(W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m} - \frac{W n_0}{g_I}\big) + \lambda \sum_{i=1}^{I} \frac{\rho_{\mathrm{L}}}{(T_i^{\max})^2}(S_i^{\text{tot}} - s_i)^3$$

subject to:

$$W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m} - \frac{W n_0}{g_I} \leq P^{\max}, \tag{12}$$

$$s_i \leq \min\{\mu_{i,\mathrm{E}}(T_i^{\max} - t), C_i^{\max}, S_i^{\text{tot}}\}, \text{for } i = 1, 2, .., I, \tag{13}$$

$$\sum_{i=1}^{I} \frac{S_i^{\text{tot}} - s_i}{T_i^{\max}} \leq \mu_{\mathrm{L}}^{\max}, \tag{14}$$

$$\text{variables: } \mathbf{s} \geq 0, \ \text{ and } 0 \leq t \leq \min_{i=1,2,...,I}\{T_i^{\max}\}.$$

Constraint (13) stems from (11) and (9), and constraint (14) stems from (10) and (8). In addition, the duration $t$ is upper limited by $\min_{i=1,2,...,I}\{T_i^{\max}\}$ due to constraint (7) (otherwise, it is meaningless to execute the computation offloading). Let $t^*$ and $\mathbf{s} = [s_1^*, s_2^*, ..., s_I^*]$ denote the optimal solutions of Problem (EEO-E). We can derive $\mu_{i,\mathrm{L}}^* = \frac{S_i^{\text{tot}} - s_i^*}{T_i^{\max}}$, $i = 1, 2, ..., I$ for Problem (EEO).

Problem (EEO-E) is still a non-convex problem with respect to $\mathbf{s}$ and $t$. To address this difficulty, we adopt a vertical decomposition that leads to the following two-layered structure of the problem:

*1) (Top-layer optimization):* On the top-layer optimization, we aim at solving the following problem:

$$\begin{aligned} \text{(EEO-E-Top):} \quad & \min V_{\text{bot}}(t) \\ \text{subject to:} \quad & 0 \leq t \leq \min_{i=1,2,...,I}\{T_i\}, \end{aligned}$$

where for each given $t$, the value of $V_{\text{bot}}(t)$ is given by the optimal value of a bottom-layer optimization problem shown below.

*2) (Bottom-layer optimization):* At the bottom-layer optimization, we aim at solving the following problem:

$$\text{(EEO-E-Bot):} V_{\text{bot}}(t) = \min t\big(W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m} - \frac{W n_0}{g_I}\big) + \lambda \sum_{i=1}^{I} \frac{\rho_{\mathrm{L}}}{(T_i^{\max})^2}(S_i^{\text{tot}} - s_i)^3$$

subject to:

$$W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m} - \frac{W n_0}{g_I} \leq P^{\max}, \tag{15}$$

$$s_i \leq \min\left\{\mu_{i,\mathrm{E}}(T_i^{\max} - t), C_i^{\max}, S_i^{\text{tot}}\right\}, \ \text{ for } i = 1, 2, ..., I, \tag{16}$$

$$\sum_{i=1}^{I} \frac{S_i^{\text{tot}} - s_i}{T_i^{\max}} \leq \mu_{\mathrm{L}}^{\max}, \tag{17}$$

$$\text{variables: } \ \mathbf{s}$$

Based on the above vertical decomposition, we identify the following result.

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics

**Proposition 1:** Given $t \in [0, \min_{i=1,2,...,I}\{T_i^{\max}\}]$, the bottom-layer Subproblem (EEO-E-Bot) is a strictly convex optimization problem with respect to $\mathbf{s}$.

*Proof:* Both the objective function of Subproblem (EEO-E-Bot) and constraint (15) are strictly convex with respect to $\mathbf{s}$ when $t$ is given. In addition, constraints (16) and (17) are affine in $\mathbf{s}$. Thus, based on the convex optimization theory [35], Subproblem (EEO-E-Bot) is a strictly convex optimization problem in $\mathbf{s}$. ■

Exploiting the convexity of the bottom-layer Subproblem (EEO-E-Bot), we thus propose an efficient algorithm to solve it. The details are shown in the next subsection.

### *A. Subroutine to Solve the Bottom-Layer Subproblem*

This subsection aims at proposing an efficient algorithm for solving the bottom-layer Subproblem (EEO-E-Bot). The convexity in Proposition 1 indicates the zero-duality gap of Subproblem (EEO-E-Bot), which enables us to adopt the primal-dual approach to solve it. Let us use $\alpha$ and $\beta$ to denote the dual variables for constraints (15) and (17), respectively. In addition, we introduce the tuple of $(\overline{r}_i, \underline{r}_i)$ to denote the dual variables for constraint $0 \leq r_i \leq Z_i$ (with $Z_i = \min\left\{\mu_{i,\mathrm{E}}(T_i^{\max} - t), C_i^{\max}, S_i^{\mathrm{tot}}\right\}$ according to constraint (16) before). Furthermore, we denote the dual vectors $\overline{\mathbf{r}} = [\overline{r}_1, \overline{r}_2, ..., \overline{r}_I]$ and $\underline{\mathbf{r}} = [\underline{r}_1, \underline{r}_2, ..., \underline{r}_I]$. Thus, the Lagrangian function of Subproblem (EEO-E-Bot) can be expressed as:

$$
\begin{aligned}
L(\mathbf{s}, \alpha, \beta, \overline{\mathbf{r}}, \underline{\mathbf{r}}) = {} & \lambda \sum_{i=1}^{I} \frac{\rho_{\mathrm{L}}}{(T_i^{\max})^2} (S_i^{\mathrm{tot}} - s_i)^3 \\
& + t\big(W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m} - \frac{W n_0}{g_I}\big) \\
& + \alpha \left( W n_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\sum_{m=i}^{I}\frac{s_m}{t}} - \frac{W n_0}{g_I} - P^{\max} \right) \\
& + \beta \left( \sum_{i=1}^{I} \frac{S_i^{\mathrm{tot}} - s_i}{T_i^{\max}} - \mu_{\mathrm{L}}^{\max} \right) + \sum_{i=1}^{I} \overline{r}_i \left(s_i - Z_i\right) - \sum_{i=1}^{I} \underline{r}_i s_i.
\end{aligned}
$$

With $L(\mathbf{s}, \alpha, \beta, \overline{\mathbf{r}}, \underline{\mathbf{r}})$, we can determine the optimal solution of Subproblem (EEO-E-Bot) in a primal-dual approach as follows.

*1) (Solving the primal problem):* Given $(\alpha, \beta, \overline{\boldsymbol{r}}, \underline{\boldsymbol{r}})$, we can derive the partial derivative of $L(\mathbf{s}, \alpha, \beta, \overline{\mathbf{r}}, \underline{\mathbf{r}})$ with respect to each $s_i$, which is denoted by function $G_i$, as follows:

$$
\begin{aligned}
G_i = & \frac{\partial L(\mathbf{s}, \alpha, \beta, \overline{\mathbf{r}}, \underline{\mathbf{r}})}{\partial s_i} \\
= & (\ln 2) n_0 (1 + \frac{\alpha}{t}) \sum_{j=1}^{i} \left(\frac{1}{g_j} - \frac{1}{g_{j-1}}\right) 2^{\frac{1}{W}\sum_{m=j}^{I}\frac{s_m}{t}} \\
& - 3\lambda \frac{\rho_{\mathrm{L}}}{(T_i^{\max})^2}(S_i^{\mathrm{tot}} - s_i)^2 - \beta \frac{1}{T_i^{\max}} + \overline{r}_i - \underline{r}_i, \\
& \text{for } i = 1, 2, ..., I. \qquad (18)
\end{aligned}
$$

Based on (18), solving the primal problem corresponds to finding $\mathbf{s}^{\mathrm{o}} = [s_1^{\mathrm{o}}, s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}]$ such that $G_i = 0$ for $i = 1, 2, ..., I$ (here, we use the superscript "o" to denote the optimality to the primal problem). To this end, we propose Subroutine-forSi which works as follows.

- *(Iterative calculations for $[s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}]$)*: Given the value of $s_1$, we can exploit the structural property of (18) to compute $s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}$ one by one. Specifically, with a given $s_1$, we can use $G_1 = 0$ to obtain the value of $\sum_{i=1}^{I} s_i$, and consequently obtain the value of $\sum_{i=2}^{I} s_i$ (i.e., given by $\sum_{i=1}^{I} s_i - s_1$). Furthermore, with the values of $\sum_{i=1}^{I} s_i$ and $\sum_{i=2}^{I} s_i$, we can use $G_2 = 0$ to obtain the value of $s_2$, and consequently obtain the value of $\sum_{i=3}^{I} s_i$ (i.e., given by $\sum_{i=2}^{I} s_i - s_2$). The above process continues until we use $G_{I-1} = 0$ to obtain the value of $s_{I-1}$, and thus the value of $s_I$. Step 3 to Step 7) in Subroutine-forSi summarize the iterative calculations, which consume $I-2$ rounds of iterations to obtain $s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}$.
- *(Line-search on $s_1$ to reach $G_I = 0$)*: Based on the above iterative calculations and the given value of $s_1$, we can obtain the values of $s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}$. Hence, we can compute the value of $G_I$ in Step 9 of Subroutine-forSi. Our objective here is to find $s_1^{\mathrm{o}}$ such that $G_I = 0$ (i.e., Step 10 to Step 12). To this end, we execute a line-search on $s_1$ with a small step-size, which is summarized by the whole WHILE-LOOP in Subroutine-forSi. Finally, with $s_1^{\mathrm{o}}$, we also obtain the corresponding $s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}$.

**Subroutine-forSi: to find $\mathbf{s}^{\mathrm{o}}$ such that $\{G_i = 0\}_{i=1,2,...,I}$**

```
1: Initialization: Set $s_1^{\text{cur.lower}} = 0$, $s_1^{\text{cur.upper}} = s_1^{\text{tot}}$. Set $\epsilon$ as a very small
   number. Set $s_1 = s_1^{\text{cur.lower}}$ and a very small step-size $\Delta$.
2: while $s_1 < s_1^{\text{cur.upper}}$ do
3:     Based on $s_1$, use $G_1 = 0$ to compute $\sum_{m=1}^{I} s_m$.
4:     Based on $s_1$, compute $\sum_{m=2}^{I} s_m = \sum_{m=1}^{I} s_m - s_1$.
5:     for $j = 2 : 1 : I - 1$ do
6:         Based on $\sum_{m=1}^{I} s_m, \sum_{m=2}^{I} s_m, ..., \sum_{m=j}^{I} s_m$, use $G_j = 0$ to
           compute $s_j$.
7:         Based on $s_j$, compute $\sum_{m=j+1}^{I} s_m = \sum_{m=j}^{I} s_m - s_j$ (notice
           that when $j = I - 1$, we can obtain $s_I$).
8:     end for
9:     Compute the value of $G_I$, based on the values of
       $\sum_{m=1}^{I} s_m, \sum_{m=2}^{I} s_m, ..., \sum_{m=I-1}^{I} s_m$ and $s_I$.
10:    if $|G_I| \leq \epsilon$ then
11:        Set $s_i^{\mathrm{o}} = s_i$ for $i = 1, 2, ..., I$ and break the whole While-Loop.
12:    end if
13:    Update $s_1 = s_1 + \Delta$.
14: end while
15: Output: $\mathbf{s}^{\mathrm{o}} = [s_1^{\mathrm{o}}, s_2^{\mathrm{o}}, ..., s_I^{\mathrm{o}}]$.
```

Until now, We can obtain $\mathbf{s}^{\mathrm{o}}$ under the given $(\alpha, \beta, \overline{\boldsymbol{r}}, \underline{\boldsymbol{r}})$. Figure 2(a) in Subsection III-C illustrates the examples of the operations of our Subroutine-forSi. We next focus on solving the dual problem.

*2) (Solving the dual problem):* With $\mathbf{s}^{\mathrm{o}}$ provided by Subroutine-forSi, we further solve the dual problem. In particular, we adopt the sub-gradient method [35] and propose the corresponding BotforDual-Algorithm for updating the dual variables (i.e., in Step 20 to Step 9) until convergence. For each given $(\alpha, \beta, \overline{\boldsymbol{r}}, \underline{\boldsymbol{r}})$, we use Subroutine-forSi to compute the corresponding $\mathbf{s}^{\mathrm{o}}$, which is used in the following dual updating. Meanwhile, in Step 4, we adopt the scheme of decreasing step-size according to $\varrho = \frac{A}{B+C*l}$ [35], where $l$ is the iteration index, and $A, B$, and $C$ are fixed parameters.

Thanks to the property of zero-duality, after convergence, BotforDual-Algorithm can output the optimal solution to the bottom-layer Subproblem (EEO-E-Bot), which is denoted by

$\mathbf{s}^{\text{bot},*} = [s_1^{\text{bot},*}, s_2^{\text{bot},*}, ..., s_I^{\text{bot},*}]$. Finally, the optimal value of the bottom-layer Subproblem (EEO-E-Bot) is given by:

$$V_{\text{bot}}(t) = \lambda \sum_{i=1}^{I} \frac{\rho_{\text{L}}}{(T_i^{\max})^2} (S_i^{\text{tot}} - s_i^{\text{bot},*})^3$$
$$+t\big(Wn_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\frac{1}{t}\sum_{m=i}^{I} s_m^{\text{bot},*}} - \frac{Wn_0}{g_I}\big). \quad (19)$$

We thus finish solving Subproblem (EEO-E-Bot).

**BotforDual-Algorithm: to solve Subproblem (EEO-E-Bot) and obtain $V_{\textbf{bot}}(t)$**

```
1: Initialization: Set $\alpha = 0$, $\beta = 0$, $\overline{\boldsymbol{r}} = 0$, and $\underline{\boldsymbol{r}} = 0$. In addition, we set $\alpha^{\text{pre}}$, $\beta^{\text{pre}}$, $\overline{\boldsymbol{r}}$ and $\underline{\boldsymbol{r}}$ as very large numbers. Set $\epsilon$ as a very small number. Set iteration index $l = 1$.
2: while $|\alpha - \alpha^{\text{pre}}| > \epsilon$ or $|\beta - \beta^{\text{pre}}| > \epsilon$ or $\left|\overline{r}_i - \overline{r}_i^{\text{pre}}\right| > \epsilon$ for any $i$ or $\left|\underline{r}_i - \underline{r}_i^{\text{pre}}\right| > \epsilon$ for any $i$ do
3:     Given the dual variables $(\alpha, \beta, \overline{\boldsymbol{r}}, \underline{\boldsymbol{r}})$, use SubforSi-Algorithm to obtain $\mathbf{s}^{\text{o}}$.
4:     Update the step-size $\varrho = \frac{A}{B+C*l}$ (which is used for updating the dual variables below).
5:     Update $\alpha^{\text{pre}} = \alpha$, $\beta^{\text{pre}} = \beta$, $\overline{\boldsymbol{r}}^{\text{pre}} = \overline{\boldsymbol{r}}$, and $\underline{\boldsymbol{r}}^{\text{pre}} = \underline{\boldsymbol{r}}$.
6:     Update $\alpha$ according to: $\alpha =$
       $\left[\alpha + \varrho \left(Wn_0 \sum_{i=1}^{I} \left(\frac{1}{g_i} - \frac{1}{g_{i-1}}\right) 2^{\frac{1}{W}\sum_{m=i}^{I}\frac{s_m^{\text{o}}}{t}} - \frac{Wn_0}{g_I} - P^{\max}\right)\right]^+$
       with function $[x]^+ = \max\{x, 0\}$.
7:     Update $\beta = \left[\beta + \varrho\left(\sum_{i=1}^{I} \frac{S_i^{\text{tot}} - s_i^{\text{o}}}{T_i^{\max}} - \mu_{\text{L}}^{\max}\right)\right]^+$.
8:     Update $\overline{r}_i = \left[\overline{r}_i + \varrho(s_i^{\text{o}} - Z_i)\right]^+$ for $i = 1, 2, ..., I$.
9:     Update $\underline{r}_i = \left[\underline{r}_i - \varrho s_i^{\text{o}}\right]^+$ for $i = 1, 2, ..., I$.
10:    Update $l = l + 1$.
11: end while
12: Set $s_i^{\text{bot},*} = s_i^{\text{o}}$ for $i = 1, 2, ..., I$.
13: Calculate $V_{\text{bot}}(t)$ according to (19).
14: Output: $\mathbf{s}^{\text{bot},*} = [s_1^{\text{bot},*}, s_2^{\text{bot},*}, ..., s_I^{\text{bot},*}]$ and $V_{\text{bot}}(t)$.
```

### B. Proposed Algorithm to Solve Top-problem (EEO-E-Top)

BotforDual-Algorithm (with Subroutine-forSi) provides $V_{\text{bot}}(t)$ for each given $t$. We then continue to solve the optimization problem on the top-layer, i.e., top-problem (EEO-E-Top) that further optimizes the NOMA-transmission duration $t$. However, the difficulty in solving the top-layer optimization (EEO-E-Top) lies in that we cannot express the value of $V_{\text{bot}}(t)$ in a closed-form expression. As a result, we cannot use conventional gradient-based approach to solve it. Fortunately, top-layer optimization (EEO-E-Top) is a single-variable optimization with the decision variable $t \in [0, \min_{i=1,2,...,I}\{T_i\}]$. As a result, we can use the line-search (LS) method to numerically find $t^*$ that can minimize $V_{\text{bot}}(t)$. The details are shown in our TopLS-Algorithm below. Notice that, for each given $t$ being evaluated, we use BotforDual-Algorithm to obtain the value of $V_{\text{bot}}(t)$ (i.e., in Step 5).

Using TopLS-Algorithm to solve the top-layer optimization (EEO-E-Top) also solves the original Problem (EEO-E). Specifically, TopLS-Algorithm outputs $t^*$, based on which we can determine $\mathbf{s}^*$ by using BotforDual-Algorithm. Then, we derive $\mu_{i,\text{L}}^* = \frac{S_i^{\text{tot}} - s_i^*}{T_i^{\max}}$ for $i = 1, 2, ..., I$ (as explained below (14) before). Finally, $(t^*, \mathbf{s}^*, \boldsymbol{\mu}_{\text{L}}^*)$ together form the optimal solutions of Problem (EEO).

The complexity of our TopLS-Algorithm can be analyzed as follows. TopLS-Algorithm executes a linear-search on $t \in [0, \min_{i\in\mathcal{K}}\{T_i\}]$. For each enumerated $t$, TopLS-Algorithm invokes BotforDual-Algorithm (in Step 5) for evaluating the value of $V_{\text{bot}}(t)$. In particular, BotforDual-Algorithm adopts the sub-gradient method to reach the dual optimum, which thus consumes the complexity of $O(\frac{1}{\epsilon^2})$ with $\epsilon$ denoting the relative error to the global optimum of the dual problem. Moreover, in each round of the iterations of BotforDual-Algorithm, in order to obtain $\mathbf{s}^{\text{o}}$ under the given tuple of the dual variables, we invoke SubforSi-Algorithm (in Step 3) which requires the complexity of $O\big(\frac{s_1^{\text{tot}}}{\Delta}(I-2)\big)$. As a result, the overall complexity of TopLS-Algorithm can be expressed as $O\big((I-2)\frac{1}{\epsilon^2}\frac{\min_{i\in\mathcal{K}}\{T_i\}s_1^{\text{tot}}}{\Delta^2}\big)$.

**TopLS-Algorithm: to solve top-layer optimization (EEO-E-Top)**

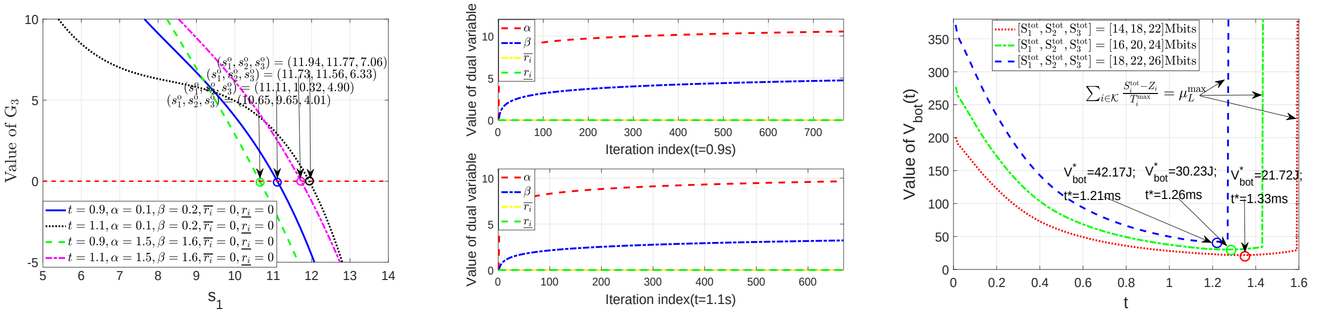
```
1: Initialization: Set step-size $\Delta$ as a very small number. Set $CBV = \infty$ and $CBS = \emptyset$.
2: Set $t = \Delta$.
3: while $t \leq \min_{i\in\mathcal{K}}\{T_i\}$ do
4:     if $\sum_{i\in\mathcal{K}} \frac{S_i^{\text{tot}} - Z_i}{T_i^{\max}} \leq \mu_{\text{L}}^{\max}$ then
5:         Use BotforDual-Algorithm to compute $V_{\text{bot}}(t)$.
6:         if $V_{\text{bot}}(t) < CBV$ then
7:’            $CBV = V_{\text{bot}}(t)$ and $CBS = t$.
8:         end if
9:     end if
10:    Update $t = t + \Delta$.
11: end while
12: Output: $t^* = CBS$ and $V_{\text{bot}}^* = CBV$.
```

### C. Numerical results for given task-ES assignment

This section shows the numerical results that validate the effectiveness of our algorithms for solving Problem (EEO-E). To this end, we set up a 3-task 3-ES scenario as follows. The 3 ESs are uniformly located on the circle with the center at $(0, 0)$ and the radius of 500m. Meanwhile, the ST is randomly located within a circular plane with the center at $(0, 0)$ and radius of 100m, and the consequent channel power gains from the ST to the ESs are generated according to the path-loss model as [37]. Based on the above settings, the random channel power gains used here are $\{g_i\}_{i\in\mathcal{I}} = \{1.205, 0.9636, 0.2365\} \times 10^{-7}$. For the ST, we set its $P^{\max} = 3$W, $\mu_{\text{L}}^{\max} = 5$Gbits/s, $W = 8$MHz, and $\rho_{\text{L}} = 0.1$W. For the three ESs, we set $[\mu_{1,\text{E}}, \mu_{2,\text{E}}, \mu_{3,\text{E}}] = [14, 12, 10]$Gbits/s and set $C_i^{\max} = 30$Mbits for each ES. For the tasks, we set $[T_1^{\max}, T_2^{\max}, T_3^{\max}] = [2, 3, 4]$ms, and $[S_1^{\text{tot}}, S_2^{\text{tot}}, S_3^{\text{tot}}] = [16, 20, 24]$Mbits. Finally, we set the parameter $\lambda = 1$ in the objective function, which indicates the equal emphasis on the energy consumption for the local computing and that for the NOMA transmission.

Figure 2(a) illustrates the line-search on $s_1$ in Subroutine-forSi for solving the primal-problem (i.e., to find $\mathbf{s}^{\text{o}}$). Figure 2(b) shows the convergence of our BotforDual-Algorithm for solving the dual problem (i.e., to find $\mathbf{s}^*$ and $V_{\text{bot}}(t)$). Figure 2(c) illustrates the rationale of our TopLS-Algorithm that executes a line-search over $t$. It is reasonable to observe that neither a too small $t$ nor a too large $t$ will be beneficial, and we need to find the optimal $t^*$ that can minimize the ST's total energy consumption.

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics



(a) Illustration of Subroutine-forSi

(b) Convergence of BotforDual-Algorithm

(c) $V_{\text{bot}}(t)$ versus $t$ under $(\mu_{1,\text{E}}, \mu_{2,\text{E}}, \mu_{3,\text{E}}) = (14, 12, 10)$.

Fig. 2: Illustration of Proposed Algorithm for Solving Problem (EEO-E).

Figure 3 demonstrates the accuracy and efficiency of our algorithm for solving Problem (EEO). Figure 3(a) validates the accuracy of our algorithm by comparing with the global-solver of LINGO (which is a commercial optimization software [36]). We test four different cases of $[\mu_{1,\text{E}}, \mu_{2,\text{E}}, \mu_{3,\text{E}}]$, and for each case, we further vary each task's computation-requirement. The results in Figure 3(a) show that our algorithm can achieve the solution almost same as the solution provided by LINGO under all the tested cases, thus validating the accuracy of our algorithm. Figure 3(b) shows the computation-time used by our algorithm and that used by LINGO's global-solver[2]. The results demonstrate that our algorithm always consumes a significantly less computation-time than the global-solver of LINGO, which validates the efficiency of our algorithm[3]. Therefore, our proposed algorithm is applicable for the practical scenarios when the ST's channel condition (for its NOMA transmission) and its computation-requirement are relatively stable within a certain period of interests.

## IV. Proposed Algorithm for Task-ES Assignment

### A. Problem formulation and algorithm design

Section II and Section III focus on minimizing the ST's total energy consumption under the given task-ES assignment as shown in Figure 1(b). In this section, we continue to investigate how to properly assign the tasks to different ESs, with the objective of further minimizing the ST's total energy consumption. Based on the previous model in Section II and the proposed algorithms in Section III, finding the optimal task-ES assignment can be regarded as finding *the optimal ordering of the tasks in $\mathcal{K}$*, as shown in Figure 1(c). Specifically, let $\pi$ denote an ordering of the tasks in $\mathcal{K}$, and further let $\pi(k)$ denote the index of task $k$ after the ordering-operation. With the ordering $\pi$, we consider that part of the workload of task $k$ will be offloaded to ES $\pi(k)$ for processing. In Figure 1(c), we show a detailed example of the ordering $\pi$ for $\mathcal{K} = \{1, 2, 3\}$ with $\pi(1) = 3$, $\pi(2) = 2$, and $\pi(3) = 1$. Mathematically, we formulate the following problem to determine the optimal ordering of the tasks ("ORP" refers to the optimal ordering problem).

$$\begin{aligned} \text{(ORP):} \quad & \min E_{(\pi)} \\ \text{subject to:} \quad & \pi(k) \neq \pi(k'), \forall k \neq k' \in \mathcal{K}, && (20) \\ & \widetilde{\pi}(j) \neq \widetilde{\pi}(j'), \forall j \neq j' \in \mathcal{K}, && (21) \\ \text{variable:} \quad & \pi. \end{aligned}$$

In Problem (ORP), the objective function represents the ST's minimum energy consumption under the ordering $\pi$, which is denoted by $E_{(\pi)}$. Notice that given the ordering $\pi$, we can use our proposed algorithms in Section III to compute the value of $E_{(\pi)}$. Constraint (20) ensures that two different tasks in $\mathcal{K}$ must be indexed differently after the ordering. We introduce $\widetilde{\pi}(j)$ to denote the original index of the task in $\mathcal{K}$, whose index is $j$ after the ordering. Constraint (21) ensures that for two different indices after the ordering, their respectively original indices (before the ordering) are different.

Directly solving Problem (ORP) is challenging, since Problem (ORP) can be regarded as an optimal matching problem with externality [38], and moreover, the reward (i.e., the ST's total energy consumption in our problem) is non-transferable among different pairs. To address this difficulty, we propose an index-swapping based algorithm (IS-Algorithm). Specifically, we first introduce the operations of index-swapping as follows.

*Definition 1: (Operations of index-swapping):* With a given ordering $\pi$, the index-swapping operation $\sigma_\pi(k, k')$ for $k \neq k' \in \mathcal{K}$ yields an updated ordering $\pi^{\text{upd}}$ as follows. Let $\pi(k) = i$ and $\pi(k') = i'$. The the updated ordering $\pi^{\text{upd}}$ is same as $\pi$, except that $\pi^{\text{upd}}(k) = i'$ and $\pi^{\text{upd}}(k') = i$.

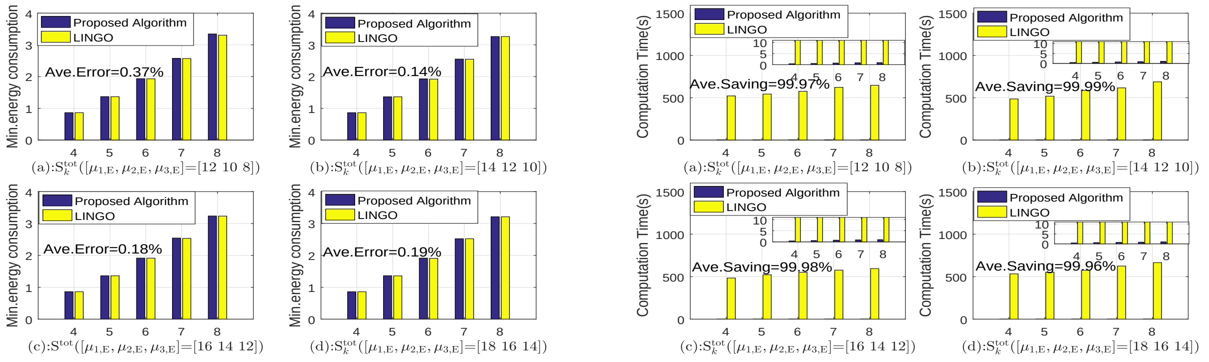
Our IS-Algorithm works in an iterative manner as follows.

- In each round of iteration, we select two different tasks $k$ and $k'$ from $\mathcal{K}$ and invoke the index-swapping operation $\sigma_\pi(k, k')$ (i.e., in Step 3 and Step 4). If this index-swapping operation can yield a more beneficial ordering, i.e., $E_{(\pi^{\text{upd}})} \leq E_{(\pi)}$, we then accept the updated ordering $\pi^{\text{upd}}$ based on $\sigma_\pi(k, k')$ in Step 7. Notice that given $\pi^{\text{upd}}$ (or $\pi$), we can use TopLS-Algorithm to compute $E_{(\pi^{\text{upd}})}$ (or $E_{(\pi)}$) in Step 5.
- To avoiding being trapped at a local optimum, we adopt the idea of the Simulated Annealing (SA), which enables us to accept a non-beneficial index-swapping operation

[2] All results are obtained with Intel(R) Core(TM) i7-7700HQ CPU at 2.80GHz.

[3] Due to the limited space, we do not show the similar numerical results for the 5-ES 5-task scenario and the 7-ES 7-task scenario, which again validate the advantage of our algorithm as Figure 3 here.

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics



(a) Accuracy of Proposed Algorithm

(b) Efficiency of Proposed Algorithm

Fig. 3: Illustration of accuracy and efficiency of our algorithm in comparison with LINGO

and the resulting updated ordering with a certain probability (i.e., in Step 10). The probability for accepting a non-beneficial index-swapping depends on both the current temperature $T^{\text{cur}}$ and the degradation in the objective, i.e., the value of gap evaluated in Step 9. When $T^{\text{cur}}$ decreases, the probability of accepting a non-beneficial swapping gradually decreases. We adopt the cooling scheduling as $T^{\text{cur}} = \frac{T^{\text{ini}}}{\ln l}$ in Step 12 (with $l$ denoting the iteration-index and $T^{\text{ini}}$ denoting the initial temperature), which can provide an asymptotic convergence [39].

**Index-Swapping Algorithm (IS-Algorithm) to find $\pi^*$**

```
1: Initialization: Initialize the ordering $\pi$ as $\pi(k) = k, \forall k \in \mathcal{K}$. Initialize $T^{\text{ini}}$, $T^{\text{cur}}$, and the iteration index $l = 1$.
2: while $T^{\text{cur}} \geq T_{\text{end}}$ do
3:     Randomly select two different $k \neq k'$ from $\mathcal{K}$.
4:     Invoke the index-swapping $\sigma_\pi(k, k')$ to yield $\pi^{\text{upd}}$.
5:     Use TopLS-Algorithm to compute the values of $E_{(\pi)}$ and $E_{(\pi^{\text{upd}})}$.
6:     if $E_{(\pi^{\text{upd}})} \leq E_{(\pi)}$ then
7:         Update $\pi = \pi^{\text{upd}}$.
8:     else
9:         Set gap $= E_{(\pi^{\text{upd}})} - E_{(\pi)}$.
10:        With probability $\exp(-\frac{\text{gap}}{T^{\text{cur}}})$, update $\pi = \pi^{\text{upd}}$ (notice that gap $>$ 0 always holds due to Step 6 before).
11:    end if
12:    Update $l = l + 1$ and $T^{\text{cur}} = \frac{T^{\text{ini}}}{\ln(l)}$.
13: end while
14: Output: $\pi^* = \pi$.
```

As a summary of all our proposed algorithms in Sections III and IV, Figure 4 shows the relationships among them.

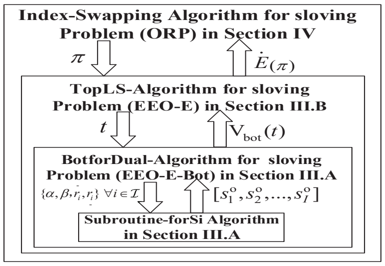

Index-Swapping Algorithm for sloving Problem (ORP) in Section IV
$\pi$
$E_{(\pi)}$
TopLS-Algorithm for sloving Problem (EEO-E) in Section III.B
$t$
$V_{\text{bot}}(t)$
BotforDual-Algorithm for sloving Problem (EEO-E-Bot) in Section III.A
$\{\alpha,\beta,r_i,r_i\}\ \forall i\in\mathcal{I}$
$[S_1^o, S_2^o, ..., S_I^o]$
Subroutine-forSi Algorithm in Section III.A

Fig. 4: Relationships among all our proposed algorithms in this work.

### B. Numerical Results

This subsection shows the performance of IS-Algorithm. We use the similar parameter-setting in Section III-C before. Figure 5 shows the convergence of our IS-Algorithm. In addition to the 3-task 3-ES scenario used before, we also set up a 5-task 5-ES scenario with the 5 ESs uniformly located on the circle with the center at $(0, 0)$ and the radius of 500m. The ST is again randomly located within a circular plane with the center at $(0, 0)$ and radius of 100m. Under this setting, the random channel power gains used here are $\{g_i\}_{i\in\mathcal{I}} = \{1.948, 1.633, 1.507, 1.095, 0.270\}\times 10^{-7}$. Meanwhile, for the 5 task, we set $[T_1^{\max}, T_2^{\max}, ..., T_5^{\max}] = [2, 3, 4, 5, 6]$ms, and $[\mu_{1,\text{E}}, \mu_{2,\text{E}}, ..., \mu_{5,\text{E}}] = [16, 14, 12, 10, 8]$Gbits/s. All the other parameters are same as those in the 3-task scenario. Figure 5 shows that our IS-Algorithm can always converge to the minimum energy consumption for each tested case.

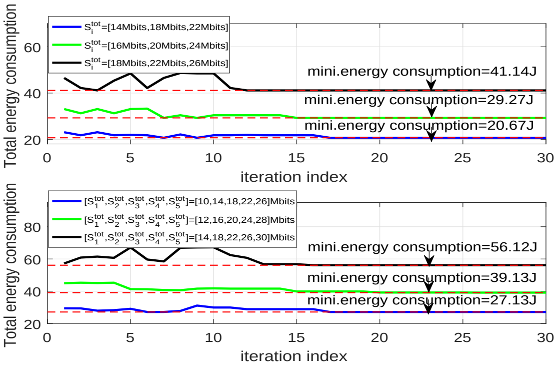


Fig. 5: Convergence of our IS-Algorithm. Top: convergence under a 3-task 3-ES case. Bottom: convergence under a 5-task 5-ES case.

Table I and Table II present the detailed results on the ordering of the ESs (obtained by our IS-Algorithm) as well as the associated optimal computation-workloads afforded by the ESs. We again use a 3-task 3-ES scenario. Table I shows the results when all tasks have the common $S_k^{\text{tot}}$ while having different delay-limits as $[T_1^{\max}, T_2^{\max}, T_3^{\max}] = [2, 3, 4]$ms. The ordering obtained by IS-Algorithm (i.e., in the second row of Table I) shows that, when we vary the common $S_k^{\text{tot}}$ for all tasks from 10Mbits and 20Mbits, Task 1 (which has the strin-

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/TII.2019.2944839, IEEE Transactions on Industrial Informatics

TABLE I: Ordering of the tasks and the accommodated workloads at different ESs with $[T_1^{\max}, T_2^{\max}, T_3^{\max}] = [2, 3, 4]$ms

| Diff. ESs | $S_k^{\text{tot}} = 10$Mbits | $S_k^{\text{tot}} = 12$Mbits | $S_k^{\text{tot}} = 14$Mbits | $S_k^{\text{tot}} = 16$Mbits | $S_k^{\text{tot}} = 18$Mbits | $S_k^{\text{tot}} = 20$Mbits |
|---|---|---|---|---|---|---|
| Ordering by IS-Algorithm | [1,2,3] | [1,2,3] | [1,2,3] | [1,2,3] | [1,2,3] | [1,2,3] |
| opt. workload by ES 1 | 7.98Mbits | 9.38Mbits | 10.64Mbits | 11.62Mbits | 12.60Mbits | 13.23Mbits |
| opt. workload by ES 2 | 7.38Mbits | 8.85Mbits | 10.18Mbits | 11.38Mbits | 12.39Mbits | 13.33Mbits |
| opt. workload by ES 3 | 4.72Mbits | 6.07Mbits | 7.25Mbits | 8.28Mbits | 9.06Mbits | 9.74Mbits |

TABLE II: Ordering of the tasks and the accommodated workloads at different ESs with $[S_1^{\text{tot}}, S_2^{\text{tot}}, S_3^{\text{tot}}] = [16, 20, 24]$Mbits

| Diff. ESs | $T_k^{\max} = 2$ms | $T_k^{\max} = 2.5$ms | $T_k^{\max} = 3$ms | $T_k^{\max} = 3.5$ms | $T_k^{\max} = 4$ms | $T_k^{\max} = 4.5$ms |
|---|---|---|---|---|---|---|
| Ordering by IS-Algorithm | [3,2,1] | [3,2,1] | [3,2,1] | [3,2,1] | [3,2,1] | [3,2,1] |
| opt. workload by ES 1 | 15.10Mbits | 18.48Mbits | 18.65Mbits | 18.76Mbits | 19.19Mbits | 19.46Mbits |
| opt. workload by ES 2 | 10.97Mbits | 14.53Mbits | 15.53Mbits | 15.90Mbits | 16.00Mbits | 15.99Mbits |
| opt. workload by ES 3 | 6.57Mbits | 9.73Mbits | 10.46Mbits | 10.47Mbits | 10.29Mbits | 10.01Mbits |

gent delay-limit) is always offloaded to ES 1 which provides the largest channel power gain and the largest computation-rate. Table II shows the results when all tasks have the common $T_k^{\text{tot}}$ while having different required computation-workloads as $[S_1^{\text{tot}}, S_2^{\text{tot}}, S_3^{\text{tot}}] = [16, 20, 24]$Mbits. The ordering obtained by our IS-Algorithm (i.e., in the 2nd row of Table II) shows that, when we vary the common $T_k^{\text{tot}}$ from 2ms to 4.5ms, Task 3 (which has the largest required computation-workload) is always offloaded to ES 1 which provides the largest channel power gain and the largest computation-rate. These results are consistent with our intuition, i.e., it will be more beneficial to offload the task of a stringent delay-limit (or large required workload) to the ESs with the large computation-rates and large channel power gains. The results in both Table I and Table II show that the ES with a larger computation-rate and larger channel gain tends to afford more offloaded workloads.

Figure 6 shows the performance advantage of our proposed NOMA-enabled offloading scheme against the conventional frequency division multiple access (FDMA) based offloading scheme. For the sake of fair comparison, we also optimize the bandwidth allocations for different ESs in the FDMA scheme. Specifically, we use a 5-ES 5-Task scenario. In the left-subplot, we consider that all tasks have the common required computation-workload $S_k^{\text{tot}}$ and vary $S_k^{\text{tot}}$ from 11Mbits to 16Mbits (we use $[T_1^{\max}, T_2^{\max}, ..., T_5^{\max}] = [2, 3, 4, 5, 6]$ms to differ the tasks). The results show that our NOMA-enabled offloading scheme can outperform the FDMA-based scheme, and the relative performance gain (i.e., the numbers marked in the figure) increases when the tasks' required computation-workloads increase. In the right-subplot, we consider that all tasks have the common delay-limit $T_k^{\max}$ and vary $T_k^{\max}$ from 3.5ms to 6ms (we use $[S_1^{\text{tot}}, S_2^{\text{tot}}, ..., S_5^{\text{tot}}] = [12, 16, 20, 24, 28]$Mbits to differ the tasks). The results again show that our NOMA-enabled offloading outperforms the FDMA-based scheme, and the relative gain (i.e., the numbers marked in the figure) increases when the delay-limit becomes more stringent.

Figure 7 shows the performance comparison between our optimal NOMA-based multi-task offloading scheme and another heuristic NOMA-based offloading scheme, in which for each task, the ST offloads a fixed portion of the computation-workload to the optimally selected ES. Meanwhile, we also optimize the NOMA-transmission duration for sending the offloaded workloads to the respective ESs as well as the local computation-rate allocations for the tasks. We use the 3-ES 3-task scenario as in Figure 3 with $[\mu_{1,\text{E}}, \mu_{2,\text{E}}, \mu_{3,\text{E}}] = [14, 12, 10]$Gbits/s. The results in Figure 7 again show that our proposed optimal NOMA-enabled scheme can always achieve the minimum energy consumption, in comparison with the heuristic NOMA-based offloading scheme.

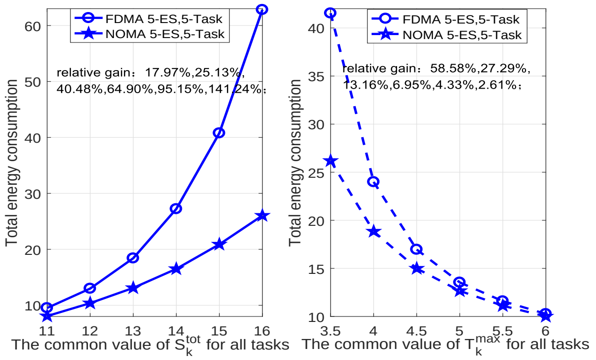


Fig. 6: Comparison between our NOMA-enabled offloading and the FDMA scheme.

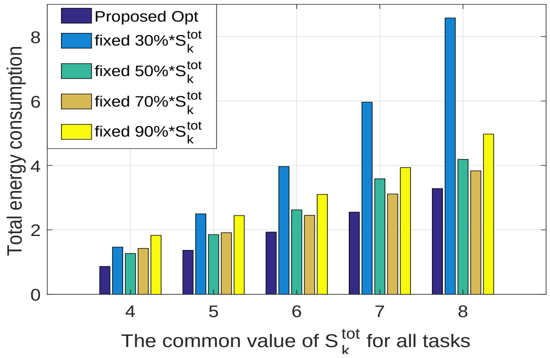


Fig. 7: Comparison between our optimal NOMA-based offloading scheme and another NOMA-based scheme with fixed computation offloading.

## V. Conclusion

In this paper, we have investigated the energy-efficient multi-task multi-access MEC via NOMA-transmission, with the objective of minimizing the ST's total energy for completing all tasks. We have adopted a two-step approach to study this problem. In the first step, we have considered a given task-ES assignment and formulated a joint optimization of the tasks' computation-offloading, local computation-rate allocation, and the NOMA-transmission time allocation. Despite the non-convexity of the formulated problem, we have exploited its layered structure and proposed an efficient algorithm to compute the optimal offloading solution. Using the proposed algorithm as the basis, we next have studied the task-ES assignment for further minimizing the ST's total energy consumption (which can be regarded as an optimal ordering of the tasks) and proposed an efficient algorithm to find the ordering of the tasks. Numerical results were presented to validate our proposed algorithms and the advantage of the NOMA-assisted multi-task multi-access offloading in terms of reducing the energy consumption. Regarding the future direction, we will consider the scenario of multiple STs and investigate how to optimally divide the ESs into different subgroups for serving different STs' offloaded workloads, by using our proposed algorithm in this work.

## References

[1] S. Kekki, *et. al.*, "MEC in 5G networks," ETSI White Paper No. 28, ISBN No. 979-10-92620-22-1, First edition, June 2018.

[2] M.S. Elbamby, C. Perfecto, C.-F Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless Edge Computing With Latency and Reliability Guarantees," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717-1737, Aug. 2019.

[3] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive Wireless Video Streaming based on Edge Computing: Opportunities and Approaches," *IEEE Transactions on Services Computing*, DOI:10.1109/TSC.2018.2828426, April 2018.

[4] J. Ren, H. Guo, C. Xu and Y. Zhang, "Serving at the Edge: A Scalable IoT Architecture based on Transparent Computing," *IEEE Networks*, vol. 31, no. 5, pp. 96-105, August 2017.

[5] J. Ni, X. Lin, and X. Shen, "Towards Edge-Assisted Internet of Things: from Security and Efficiency Perspectives," *IEEE Network*, vol. 33, no. 2, pp. 50-57, March/April 2019.

[6] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-Ground Integrated Mobile Edge Networks: Architecture, Challenges and Opportunities," *IEEE Communication Magazine*, vol. 56, no. 8, pp. 26-32, 2018.

[7] F. Giust, X. Costa-Perez, and A. Reznik, "Multi-Access Edge Computing: An Overview of ETSI MEC ISG," *IEEE 5G Tech Focus*, vol. 1, no. 4, Dec. 2017.

[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S, Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657-1681, third-quarter 2017.

[9] H. Guo, J. Liu, and J. Zhang, "Computation Offloading for Multi-Access Mobile Edge Computing in Ultra-Dense Networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 14-19, Aug. 2018.

[10] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient Multi-user Computation Offloading for Mobile-edge Cloud Computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.

[11] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks With Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 19324-19337, March 2018.

[12] M.H. Chen, B. Liang, and M. Dong, "Multi-User Multi-Task Offloading and Resource Allocation in Mobile Cloud Systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6790-6805, Oct. 2018.

[13] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized Computation Offloading Performance in Virtual Edge Computing Systems via Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4005-4018, June 2019.

[14] L. Huang, S. Bi, and Y.J. Zhang, "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks," *IEEE Trans. on Mobile Computing*, DOI:10.1109/TMC.2019.2928811, July 2019.

[15] T.Q. Dinh, Q.D. La, T.Q.S. Quek, and H. Shin, "Learning for Computation Offloading in Mobile Edge Computing," *IEEE Trans. on Communications*, vol. 66, no. 12, pp. 6353-6367, Dec. 2018.

[16] Y. Zhang, X. Lan, Y. Li, L. Cai, and J. Pan, "Efficient Computation Resource Management in Mobile Edge-Cloud Computing," *IEEE Internet of Things Journal*, vol. 6, no.2, pp.3455-3466, April 2019.

[17] D. Zhang, L. Tan. J. Ren, M.K. Awad, S. Zhang, Y. Zhang, P.J. Wan, "Near-optimal and Truthful Online Auction for Computation Offloading in Green Edge-Computing Systems," *IEEE Transactions on Mobile Computing*, DOI:10.1109/TMC.2019.2901474, Feb. 2019.

[18] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial Offloading Scheduling and Power Allocation for Mobile Edge Computing Systems," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6774-6785, Aug. 2019.

[19] S. Bi and Y.J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile-edge Computing with Binary Computation Offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177-4190, June 2018.

[20] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double Auction-Based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things," *IEEE Trans. on Industrial Informatics*, vol. 14, no. 10, pp. 4692-4701, Oct. 2018.

[21] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A Blockchain-based Non-Repudiation Network Computing Service Scheme for Industrial IoT," *IEEE Trans. on Industrial Informatics*, DOI:10.1109/TII.2019.2897133, Feb. 2019.

[22] X. Peng, J. Ren, L. She, D. Zhang, J. Li, and Y. Zhang, "BOAT: A Block-Streaming App Execution Scheme for Lightweight IoT Devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1816-1829, June 2018.

[23] J. Liu. G. Shou, Y. Liu, Y. Hu, and Z. Guo, "Performance Evaluation of Integrated Multi-Access Edge Computing and Fiber-Wireless Access Networks," *IEEE Access*, vol. 6, pp. 30269-30279, May 2018.

[24] S. Sardellitti, M. Merluzzi, and S. Barbarossa, "Optimal Association of Mobile Users to Multi-Access Edge Computing Resources," in *Proc. of IEEE ICC'2018 Workshops*.

[25] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C.L. I, and H. Poor, "Application of Non-Orthogonal Multiple Access in LTE and 5G Networks," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 185-191, Feb. 2017.

[26] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Non-orthogonal Multiple Access for 5G and Beyond," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2347-2381, Oct. 2017.

[27] D. Zhai, R. Zhang, L. Cai, and F.R. Yu, "Delay Minimization for Massive Internet of Things with Non-Orthogonal Multiple Access," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 553-566, June 2019.

[28] J. Zhu, J. Wang, Y. Huang, S. He, X. You, and L. Yang, "On Optimal Power Allocation for Downlink Non-Orthogonal Multiple Access Systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 12, pp. 2744-2757, Dec. 2017.

[29] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-Efficient User Scheduling and Power Allocation for NOMA based Wireless Networks with Massive IoT Devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1857-1868, June 2018.

[30] Y. Wu, L. Qian, H. Mao, X. Yang, H. Zhou, and X. Shen, "Optimal Power Allocation and Scheduling for Non-Orthogonal Multiple Access Relay-Assisted Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2591-2606, Nov. 2018.

[31] A. Kiani, N. Ansari, "Edge Computing Aware NOMA for 5G Networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1299-1306, April, 2018.

[32] Z. Ding, D.W.K. Ng, R. Schober, and H.V. Poor, "Delay Minimization for NOMA-MEC Offloading," *IEEE Signal Processing Letters*, vol. 25, no. 12, Dec. 2018.

[33] Y. Wu, L. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-Minimization Nonorthogonal Multiple Access enabled Multi-User Mobile Edge Computation Offloading," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 392-407, June 2019.

[34] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D.O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569-4581, Sept. 2013.

[35] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] L. Schrage, Optimization Modeling with LINGO (the 5th edition). Lindo System, Jan. 1999.

[37] R. Zhang, "Optimal Dynamic Resource Allocation for Multi-antenna Broadcasting with Heterogeneous Delay-Constrained Traffic," *IEEE Jour-*

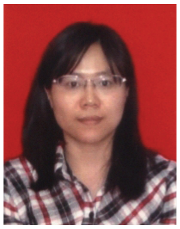
*nal on Selected Topics on Signal Processing*, vol. 2, no. 2, pp. 243-255, Apr. 2008.

[38] R. Massin, C.J. Le Martret, P. Cibalt, “A Coalition Formation Game for Distributed Node Clustering in Mobile Ad Hoc Networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3940-3952, June 2017.

[39] E. Talbi, Metaheuristics from Design to Implementation, John Wiley & Sons Ltd, 2009.

**Yuan Wu (S’08-M’10-SM’16)** received the Ph.D degree in Electronic and Computer Engineering from the Hong Kong University of Science and Technology, Hong Kong, China, in 2010. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China. Prior to that, he was a Full Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests focus on resource management for wireless networks, green communications and computing, and smart grid. Dr. Wu was a recipient of the Best Paper Award of the IEEE International Conference on Communications in 2016.

**Binghua Shi** is currently pursuing his M.S. degree in College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interest focuses on resource management for wireless networks, non-orthogonal multiple access and mobile edge computing.



**Li Ping Qian (S’08-M’10-SM’16)** received the PhD degree in Information Engineering from the Chinese University of Hong Hong, Hong Kong, in 2010. She worked as a postdoctoral research associate at the Chinese University of Hong Kong, Hong Kong, during 2010-2011. Since 2011, she has been with College of Information Engineering, Zhejiang University of Technology, China, where she is currently a full Professor. From 2016 to 2017, she was a visiting scholar with the Broadband Communications Research Group, ECE Department, University of Waterloo. Her research interests include wireless communication and networking, resource management in wireless networks, massive IoTs, mobile edge computing, emerging multiple access techniques, and machine learning oriented towards wireless communications. She was a co-recipient of the IEEE Marconi Prize Paper Award in Wireless Communications in 2011, the Best Paper Award from IEEE ICC 2016, and the Best Paper Award from IEEE Communication Society GCCTC 2017. She is currently on the Editorial Board of IET Communications.

**Fen Hou** is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Macau. She received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2008. She worked as a postdoctoral fellow in the Electrical and Computer Engineering at the University of Waterloo and in the Department of Information Engineering at the Chinese University of Hong Kong from 2008 to 2009 and from 2009 to 2011, respectively. Her research interests include resource allocation and scheduling in broadband wireless networks, protocol design and QoS provisioning for multimedia communications in broadband wireless networks, Mechanism design and optimal user behavior in mobile crowd sensing networks and mobile data offloading. She is the recipient of IEEE GLOBECOM Best Paper Award in 2010 and the Distinguished Service Award in IEEE MMTC in 2011.

**Jiali Cai** received the Bachelor degree in Communication Engineering from College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2019. Her final year project focuses on multi-access mobile edge computing via non-orthogonal multiple access.

**Xuemin (Sherman) Shen (M’97-SM’02-F’09)** is a University Professor and the Associate Chair for Graduate Studies, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen’s research focuses on wireless resource management, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is the IEEE ComSoc VP Publication, was an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom’16, Infocom’14, IEEE VTC’10 Fall, and Globecom’07, the Symposia Chair for IEEE ICC’10, the Tutorial Chair for IEEE VTC’11 Spring and IEEE ICC’08, the General Co-Chair for ACM Mobihoc’15, and the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Internet of Things Journal, and IEEE Network, a Founding Area Editor for IEEE Transactions on Wireless Communications; and an Associate Editor for IEEE Transactions on Vehicular Technology and IEEE Wireless Communications, etc. Dr. Shen received the IEEE ComSoc Education Award, the Joseph LoCicero Award for Exemplary Service to Publications, the Excellent Graduate Supervision Award in 2006, and the Premier’s Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.