

# Learning-Assisted Optimization for Energy-Efficient Scheduling in Deadline-Aware NOMA Systems

Lei Lei<sup>1</sup>, Member, IEEE, Lei You<sup>2</sup>, Student Member, IEEE, Qing He, Member, IEEE,  
Thang X. Vu, Member, IEEE, Symeon Chatzinotas, Senior Member, IEEE,  
Di Yuan, Senior Member, IEEE, and Björn Ottersten<sup>3</sup>, Fellow, IEEE

**Abstract**—In this paper, we study a class of minimum-energy scheduling problems in non-orthogonal multiple access (NOMA) systems. NOMA is adopted to enable efficient channel utilization and interference mitigation, such that base stations can consume minimal energy to empty their queued data in presence of transmission deadlines, and each user can obtain all the requested data timely. Due to the high computational complexity in resource scheduling and the stringent execution-time constraints in practical systems, providing a time-efficient and high-quality solution to 5G real-time systems is challenging. The conventional iterative optimization approaches may exhibit their limitations in supporting online optimization. We herein explore a viable alternative and develop a learning-assisted optimization framework to improve the computational efficiency while retaining competitive energy-saving performance. The idea is to use deep-learning-based predictions to accelerate the optimization process in conventional optimization methods for tackling the NOMA resource scheduling problems. In numerical studies, the proposed optimization framework demonstrates high computational efficiency. Its computational time is insensitive to the input size. The framework is able to provide optimal solutions as long as the learning-based predictions satisfy a derived optimality condition. For the general cases with imperfect predictions, the algorithmic solution is error-tolerable and performance scalable, leading the energy-saving performance close to the global optimum.

**Index Terms**—Non-orthogonal multiple access, deep neural network, energy optimization, resource scheduling.

## I. INTRODUCTION

THE UPCOMING 5G communication system is envisioned to provide low-latency services with low-energy

consumption in supporting users' high data demand [1]. According to Cisco's annual visual network index reports [2], numerous users in the system will be increasingly hungry for large amounts of data, e.g., high-definition video streaming, and meanwhile they will be also highly demanding on timely services, in which the users are able to obtain all the requested data as soon as possible without unacceptable delay. Reducing the duration of data transmission and squeezing each terminal's waiting time in data service, are important for measuring the overall quality-of-experience of the users [1].

Non-orthogonal multiple access (NOMA), as one of the promising techniques for 5G new radio, is currently under investigation for improving the system performance in throughput, fairness, latency, and energy consumption [3]–[8]. On the one hand, the performance gains of NOMA over orthogonal multiple access schemes are significant, mainly due to its high efficiency in channel utilization and the application of successive interference cancellation (SIC) for co-channel interference mitigation. On the other hand, the high computational complexity in resource allocation, and the stringent execution-time requirements in real-time networks, may pose obstacles in applying NOMA in practical systems. For example, previously proposed optimal/suboptimal solutions for solving some high-complexity NOMA resource allocation problems, may not be practical in real-time systems as they may require immense computational capabilities and time, especially for large-scale instances [3], [4], [9]. As a matter of fact, the time limit to complete a decision-making process in practice, requires to be within seconds or milliseconds [1], [10], whereas solving NOMA resource allocation problems to a satisfactory level would require a much longer span of computing time.

The optimal and suboptimal algorithmic solutions for NOMA resource allocation have been extensively investigated in the literature. The NP-hardness of NOMA resource optimization has been discussed in [4]. The majority of the proposed algorithms in the literature are either based on tailored exact algorithms or meticulously designed heuristics. The former typically leads to prohibitively high complexity, and can only be applied for small-medium instances [9]. For the latter, a trial-and-error philosophy has been widely adopted in suboptimal algorithm development [3]–[5], [7], [8], [11]. This type of design method first comes up with a heuristic idea for simplifying the optimization process, followed by intensive and time-consuming tests. To achieve competitive

Manuscript received July 5, 2018; revised December 13, 2018; accepted February 22, 2019. Date of publication March 4, 2019; date of current version August 16, 2019. The work of L. You and D. Yuan was supported by the Swedish Research Council. This work was supported in part by the European Research Council Project AGNOSTIC under Grant 742648, in part by the FNR Bilateral Project LARGOS, in part by the FNR CORE Project ROSETTA under Grant 11632107, and in part by the FNR CORE Project ProCAST under Grant R-AGR-3415-10. The associate editor coordinating the review of this paper and approving it for publication was E. Ayanoglu. (*Corresponding author: Lei Lei.*)

L. Lei, T. X. Vu, S. Chatzinotas, and B. Ottersten are with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, 1855 Luxembourg City, Luxembourg (e-mail: lei.lei@uni.lu; thang.vu@uni.lu; symeon.chatzinotas@uni.lu; bjorn.ottersten@uni.lu).

L. You and D. Yuan are with the Department of Information Technology, Uppsala University, 75237 Uppsala, Sweden (e-mail: lei.you@it.uu.se; di.yuan@it.uu.se).

Q. He is with the Department of Network and Systems Engineering, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: qhe@kth.se).

Digital Object Identifier 10.1109/TGCN.2019.2902838

performance, one has to rely on the feedback from the tested results, possibly a huge amount of data, in order to identify the flaws and issues in the original algorithm design [11]. Then adjustments and updates can be made in algorithm development. This process often has to repeat until satisfactory performance is achieved. The algorithmic solutions are typically designed as one algorithm solving one optimization problem. However, if some key aspects vary, e.g., adopting new performance metrics, constraints, or parameters, the algorithms may need to undergo revisions, as the changes can possibly destroy the problem structure, and make the original algorithm's performance degraded or even make the solution infeasible. Towards real-time resource allocation, this algorithm design approach and the previous NOMA algorithms may exhibit their limitations in supporting online resource management in highly dynamic and complex networks [3].

In general, once an optimization problem is proved to be hard to solve, it is difficult to expect that a heuristic solution can meanwhile achieve satisfactory performance and with very low complexity to support real-time optimization [12]. In most cases, a scheduler may have to make a trade-off between the algorithm's computational complexity and the solution quality, which is a dilemma in the conventional algorithm development. Thus, being aware of the shortcomings, we are motivated to explore an alternative in solution development for NOMA resource optimization. Machine learning can provide a powerful alternative in algorithm design for the complex and highly dynamic systems. As an approach in the toolbox for 5G network optimization, it has received considerable research attention recently [13]–[17]. Sun *et al.* [13] adopt a deep-learning approach and directly apply it as a heuristic to predict the power allocation for a classical interference-channel power control problem. Ghadimi *et al.* [14] considered reinforcement learning in resource management. In [15] and [16], machine/deep learning based approaches were applied in caching networks and multi-antenna systems. Tsakmalis *et al.* [17] investigated a multivariate Bayesian active learning method for centralized cognitive radio networks.

#### A. Contributions

In this paper, the main contributions are from three aspects. Firstly, we address two emerging issues in resource scheduling for practical NOMA systems. That is, by taking into account the dynamic duration in emptying queued data in NOMA, how to efficiently schedule resources to satisfy all the users' data demand within deadlines, and how to cope with the high complexity in resource scheduling, while retaining fast execution in real-time applications. We formulate two types of minimum-energy scheduling problems with a set of practical deadline constraints. Unlike previously studied wireless scheduling problems [18]–[20], the multiple deadlines for satisfying individual user's demand in NOMA systems are taken into account in this work. These make the resource scheduling problems even challenging to solve. We derive structural analysis for the problems to enable reasonable solutions, and then

provide numerical and analytical results to illustrate significant energy-saving gains of applying NOMA and our approach.

Secondly, we investigate the approach of combining deep learning with conventional optimization algorithms. We propose a learning-assisted optimization framework to leverage the power of deep learning and iterative optimization algorithms, aiming at providing efficient and competitive solutions for solving a class of scheduling problems. We embed a learning component into the algorithmic design, and let it learn the behaviors of optimal decisions by training. The learning component is then used to provide guidance to help the conventional iterative algorithm to accelerate its optimization process in real-time systems.

Thirdly, the proposed approach provides an alternative way to tackle the trade-off issue between solution quality and computational complexity. The conventional iterative algorithms may lead to low computational efficiency [4], [9], [19]–[21], and the simple heuristics typically have limitations in achieving satisfactory performance [3], [12]. These are the undesired performance for practical network optimization [10]. In addition, unlike previously proposed learning approaches, e.g., [11] and [13], simply applying machine-learning models to directly predict the variable values may not be a viable choice for the considered problems in this work due to the huge number of variables in the optimization. The proposed learning-assisted optimization can reap the benefits of machine learning and advanced optimization methods, e.g., efficient computations from the former, high-quality and feasible solution from the latter. In the proposed approach, the majority of the computational complexity has been concentrated on the offline data generation and training phase. The after-training phase is computationally light, and its computational time is insensitive to the input size, which is promising to enable an online optimization in real-time systems.

The rest of the paper is organized as follows. Section II presents the system models for NOMA, group scheduling, data transmission. Section III formulates two NOMA based minimum-energy scheduling problems with various types of deadline constraints, and provides structural analysis. Section IV analyzes the two problems' tractability and proposes a learning-assisted algorithmic framework. Numerical results are demonstrated in Section V. Conclusions are given in Section VI.

## II. SYSTEM MODEL

We consider a downlink NOMA system consisting of one base station (BS) serving  $K$  mobile users in a common channel with bandwidth  $B$ . The user set is denoted as  $\mathcal{K} = \{1, \dots, k, \dots, K\}$ . We define the starting point of a schedule as  $T_0$ . With the time constraint at the transmitter side, the BS needs to empty its queued data demand  $D_1, \dots, D_K$  within a time limit  $T_{tot}$ , counting from  $T_0$  until all the users' data demands  $D_1, \dots, D_K$  have been delivered. With the time requirements at the receiver side, to improve users' satisfaction, each user  $k$  in the system has a specific requirement  $T_k$  representing the user's tolerable waiting time to obtain all the requested data, counting from  $T_0$  until user  $k$ 's data  $D_k$  has been successfully transmitted. Considering various

requirements of quality-of-service (QoS) and the performance metric, the BS is required to optimize its scheduling strategy in data transmission, in order to satisfy all users' data demand and to meet a variety of deadlines. Throughout the paper, we refer to  $T_{tot}$  as the total transmission deadline, and define  $T_1, \dots, T_K$  as the individual transmission deadlines for user  $1, \dots, K$ , respectively. We remark that the users' traffic is assumed static during one decision-making period. When new traffic arrives during this period, the request will be formed as a new snapshot, and will be handled in the next decision-making procedure. The goal of this work is to significantly reduce the decision-making delay by the proposed learning-assisted optimization approach, such that the dynamic traffic can be addressed timely.

#### A. User-Grouping Scheduling and NOMA

Towards flexible and efficient data transmission, we adopt dynamic user grouping and apply NOMA in each scheduled group to mitigate co-channel interference and achieve efficient frequency utilization. Each mobile device is capable of performing multi-user detection and iterative SIC in NOMA [5]. We refer to a cluster  $u$  as a user group/set, consisting of one or multiple users [19]. Let  $\mathcal{K}_u$  be the set of users in group  $u$ . Once a group  $u$  is scheduled, the BS will transmit data to all the users in  $\mathcal{K}_u$  with positive rates and last for a certain duration  $t_u$ . In practice, the decision-making for grouping and scheduling occurs at the beginning of each scheduling frame. The BS collects users' channel state information (CSI) and the data requests at every frame. Based on the collected information, the BS makes decisions for grouping and scheduling. Then the BS sends the control signal (before data transmission) to the users to notify that each individual user should receive data in which time slots.

Enumerating all the combinations of the user groups, provides  $2^K - 1$  possible candidates in total. The union of all the groups is denoted by  $\mathcal{U} = \{1, \dots, u, \dots, U\}$ , where  $U = 2^K - 1$ . For example, suppose  $\mathcal{K} = \{1, 2, 3\}$ , by enumeration there are seven candidate groups  $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ , i.e.,  $U = 7$ . These user groups can be selectively scheduled in a sequential manner to deliver users' data demand, and each user's demand can be flexibly delivered in multiple groups with non-uniform data rates. The cardinality of  $\mathcal{U}$  increases exponentially with  $K$ . To reduce the complexity, some simple schemes are adopted in previous works, e.g., time division multiple access (TDMA) which can be simply enabled by setting  $U = K$  and  $|\mathcal{K}_u| = 1$  in group enumeration. We remark that the simple scheduling schemes may either fail to satisfy all users' QoS or lead to performance degradation. For example, adopting TDMA to serve each user, one at a time, may fail to meet strict transmission deadlines [21]. Thus in this work, the selection of scheduled user groups is not predefined, but is subject to optimization.

Within a group, NOMA is applied to alleviate the intra-group interference. In NOMA, the BS transmits a superposed signal  $x_u$  to all the users in  $\mathcal{K}_u$ ,

$$x_u = \sum_{k \in \mathcal{K}_u} \sqrt{p_k} s_k \quad (1)$$

where  $p_k$  is the transmit power for user  $k$  and  $s_k$  is the signal intended for user  $k$ . Note that the transmit power for each user may not necessarily be uniform. It can be pre-optimized according to the channel conditions and subject to practical power constraints and the NOMA protocol. For example, a low-complexity scheme "fractional transmit power control (FTPC)" [22] can be adopted for power allocation among users.

The received signal  $y_k^u$  at user  $k$ 's receiver reads,

$$y_k^u = \sqrt{p_k} h_k s_k + \sum_{j \in \mathcal{K}_u \setminus \{k\}} \sqrt{p_j} h_k s_j + n \quad (2)$$

where  $h_k$  is the channel coefficient from the BS to user  $k$ , and  $n$  is the additive white Gaussian noise (AWGN) with zero mean and variance  $\sigma^2$ . According to the basis of NOMA, we adopt the descending order of channel gains as the decoding order [3]. Note that throughout the paper, for the convenience in presentation, we assume that the descending order of channel gains is consistent with the user index, i.e.,  $\frac{|h_1|^2}{\sigma^2} \geq \frac{|h_2|^2}{\sigma^2} \geq \dots \geq \frac{|h_K|^2}{\sigma^2}$ . By applying SIC, the signal-to-interference-plus-noise ratio (SINR) for user  $k$  in a group  $u$ , say  $\mathcal{K}_u = \{1, \dots, k, \dots, K\}$ , is expressed as,

$$\text{SINR}_k^u = \frac{p_k |h_k|^2}{\sum_{j=1}^{k-1} p_j |h_k|^2 + \sigma^2} \quad (3)$$

The corresponding data rate of user  $k$  in  $u$  is

$$r_k^u = B \log(1 + \text{SINR}_k^u) \quad (4)$$

In (3), the user  $k$ 's receiver can decode and remove the interfering signals of users  $k+1, \dots, K$ . The signals intended for users  $1, \dots, k-1$  are treated as interference [3]. We use a simple example to illustrate the principle and the assumption behind this. Considering a two-user group  $u'$  with  $\mathcal{K}_{u'} = \{k, k+1\}$  and  $|h_k|^2 \geq |h_{k+1}|^2$ . The system assumes that if the weak user  $k+1$  is able to decode the desired signal from its receiver, i.e.,  $\text{SINR}_{k+1}^{u'} = \frac{p_{k+1} |h_{k+1}|^2}{p_k |h_{k+1}|^2 + \sigma^2} \geq \gamma$ , where  $\gamma$  is a threshold, then the strong user  $k$  can also decode and remove this signal due to  $\frac{p_{k+1} |h_k|^2}{p_k |h_k|^2 + \sigma^2} \geq \frac{p_{k+1} |h_{k+1}|^2}{p_k |h_{k+1}|^2 + \sigma^2} \geq \gamma$ .

#### B. Duration of Data Transmission

In scheduling, we consider dynamic duration in data transmission which comes from draining the queued data at the BS. To properly measure the time span, the data transmission at the BS is considered being continuous in the paper. This is supported by multiple stop-and-wait processes in LTE systems [10]. We define the total transmission duration as  $\tau_{tot} = \sum_{u \in \mathcal{U}} t_u$ , the simple summation of the transmission duration of all the scheduled groups.

During  $\tau_{tot}$ , each individual user's time consumption contains two parts, data transmission and waiting in the queue. We use  $\tau_k$  to represent the time consumption that the BS spends to deliver all the requested data  $D_k$  to user  $k$ . When a user group  $u$  is scheduled, the time resource is exclusive for the users in  $\mathcal{K}_u$ , and their active transmission time in data services is  $t_u$ . On the other hand,  $t_u$  is also counted as the queuing time for those users in  $\mathcal{K} \setminus \mathcal{K}_u$ , waiting in the transmitter's

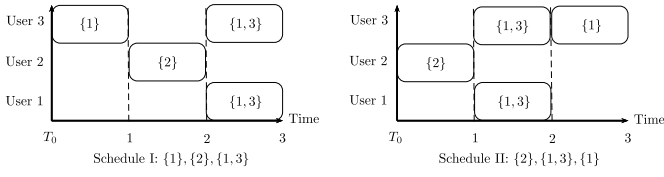


Fig. 1. An illustrative example for two scheduling schemes and their time lines in data transmission. Users' time deadlines are  $T_1 = T_3 = 3$  and  $T_2 = 1$ . In Schedule I, user 2 fails to meet its individual deadline. By optimizing the order in Schedule II, all the users can obtain their data demand  $D_1$ ,  $D_2$ , and  $D_3$  in time.

cache until their queued data become scheduled and emptied. Thus,  $\tau_k$  not only depends on the duration of the scheduled groups, but is also relevant to the scheduling order. Given  $n$  scheduled groups with a specific order, e.g.,  $u = 1, 2, \dots, n$ , the total time span for serving a user  $k$  is defined as  $\sum_{u=1}^{i(k)} t_u$ , where  $i(k)$  is the position of the last group used to serve user  $k$  in the order. We use a 3-user example in Fig. 1 to illustrate the insights of the scheduling order. To satisfy users' demand  $D_1$ ,  $D_2$ , and  $D_3$  within their individual deadlines  $T_1 = 3$ ,  $T_2 = 1$ , and  $T_3 = 3$ , from time  $T_0$  three user groups  $u_1$ ,  $u_2$ ,  $u_3$  with  $\mathcal{K}_{u_1} = \{1\}$ ,  $\mathcal{K}_{u_2} = \{2\}$ , and  $\mathcal{K}_{u_3} = \{1, 3\}$  are sequentially used to transmit data with the same normalized duration  $t_{u_1} = t_{u_2} = t_{u_3} = 1$ . For both Schedules I and II, the total time consumption is  $\tau_{tot} = 3$ . In Schedule I, for user 1 and user 3, the time span is 3 time units to deliver their requested data, then  $\tau_1 = \tau_3 = 3$ , while  $\tau_2 = 2 > T_2$ , thus exceeding the deadline of user 2. Keeping the same active groups but changing the order in Schedule II, data transmission for individual users can be completed more efficiently than Schedule I with  $\tau_1 = 3$ ,  $\tau_2 = 1$ , and  $\tau_3 = 2$ .

### III. PROBLEM FORMULATION AND STRUCTURAL ANALYSIS

We consider two NOMA based energy-efficient scheduling problems (NESP) in this section, in order to transmit all the users' demands with minimal energy consumption, and meanwhile to meet deadline constraints.

#### A. NESP With the Transmitter's Deadline

The first NESP problem is formulated in P1. It is focused on the transmitter perspective, which is motivated by the fact that the BS is expected to empty the queued data within a limited time interval, such that the occupied time-frequency resources can be released for serving the upcoming demand. In modeling the problem, firstly we enumerate all the possible user groups  $K_1, \dots, K_u, \dots, K_U$ . Each group is associated with a variable, i.e., variables  $t_1, \dots, t_u, \dots, t_U$  for groups  $K_1, \dots, K_u, \dots, K_U$ , respectively. The optimization process determines which groups to be scheduled to transmit users' demand and for how long. If  $t_u$  is positive, it means group  $u$  is activated to transmit data with a certain duration  $t_u$ , otherwise the group will not be scheduled.

$$P1: \min_{t_u} \sum_{u \in \mathcal{U}} t_u \sum_{k \in \mathcal{K}_u} p_k \quad (5a)$$

$$\text{s.t.} \quad \sum_{u \in \mathcal{G}_k} t_u r_k^u \geq D_k, \quad \forall k \in \mathcal{K} \quad (5b)$$

$$\sum_{u \in \mathcal{U}} t_u \leq T_{tot} \quad (5c)$$

The objective (5a) is to minimize the energy consumption in data transmission, where  $\sum_{k \in \mathcal{K}_u} p_k$  is the total power consumption in group  $u$ . In general, the groups with larger cardinality lead to higher sum power. By constraints (5b), the requested data for each user  $k$  must be delivered, where set  $\mathcal{G}_k = \{u \in \mathcal{U} : k \in \mathcal{K}_u\}$  contains all the groups that user  $k$  is included, and  $r_k^u$  is defined by equation (4). In (5c), the BS should complete all the data delivery within deadline  $T_{tot}$ .

*Remark 1:* The SIC process in NOMA can impose extra decoding delay at the receivers. For dealing with this issue in delay-sensitive scenarios, P1 can be further enhanced by introducing extra constraints to confine the delay in receivers' signal processing. In practice, iterative SIC which employs multiple iterations in signal processing is adopted in NOMA receivers [5], [7]. If a user's signal decoding fails in the first iteration, in the subsequent iterations it will be decoded again [23]. With soft-decision in iterative SIC, as the number of iterations increases, it can result in longer processing delay. Thus we can confine the maximum number of SIC iterations to avoid long process delay. When a user group  $u$  is scheduled, the required iterations in general linearly increase with the number of users in the group [5], [7]. Let  $L$  be the iterations required for single-user decoding ( $|\mathcal{K}_u| = 1$ ). The time spent at an iteration is  $\tau$ . The required SIC iterations for a multi-user group  $u$  are  $\min(L|\mathcal{K}_u|, L_{max})$ , where  $L_{max}$  is the maximum decoding iterations for each receiver. The accumulated processing delay for user  $k \in \mathcal{K}_u$  in group  $u$  is defined as  $t_u \times r_k^u \times \min(L|\mathcal{K}_u|, L_{max}) \times \tau$  which increases with the transmitted data bits  $t_u r_k^u$  and the number of iterations. In P1, the following constraints  $\sum_{u \in \mathcal{G}_k} t_u r_k^u \min(L|\mathcal{K}_u|, L_{max}) \tau \leq \tau_k^{max}$ ,  $\forall k \in \mathcal{K}$  can be imposed to confine each receiver's processing delay, where  $\tau_k^{max}$  is the maximum tolerable processing delay for user  $k$ .

#### B. NESP With the Receivers' Deadlines

In the following, from the receiver's perspective, we consider the second NESP problem in P2 which puts more emphasis on users' experience. We take into account each user's transmission deadline in NESP. In practice, multiple users may request data service simultaneously with diverse deadline requirements. For instance, a user who is in a real-time video conference would have strict time requirement to obtain the requested data, while another user being in a non-real-time data download process could be more tolerable to transmission delay. The BS then needs to decide the optimal order in addition to determining optimal user groups and their duration.

$$P2: \min_{t_u} \sum_{u \in \mathcal{U}} t_u \sum_{k \in \mathcal{K}_u} p_k \quad (6a)$$

$$\text{s.t.} \quad (5b)$$

$$\sum_{u \in \mathcal{G}_1 \cup \mathcal{G}_2 \cup \dots \cup \mathcal{G}_k} t_u \leq T_k, \quad \forall k \in \mathcal{K} \quad (6b)$$

In P2, the objective function, the user-demand constraints keep same as (5a) and (5b). For the deadline constraints, instead of (5c) in P1, each user's demand in P2 must be delivered within an individual deadline in constraints (6b). Note that to avoid redundancy, we do not jointly impose the deadlines  $T_{tot}$  and  $T_1, \dots, T_K$  to P2. We remark that when multiple users have deadlines in P2, the optimization process no longer resembles the structure of P1, though all the other parts remain unchanged. This change makes the scheduling problem P2 more complicated since the consumed time for serving a user's demand cannot simply be the accumulated activation time of the user, and it must be based on a specific scheduling order. Thus deciding an optimal scheduling order is important for P2. In constraints (6b), we firstly define an scheduling order (assuming  $T_1 \leq \dots \leq T_K$  in P2), and then discuss its optimality in Lemma 1. The total time span for user  $k$  is accumulated by its transmission duration and the queuing duration in the scheduled groups in  $\mathcal{G}_1 \cup \mathcal{G}_2 \cup \dots \cup \mathcal{G}_k$ . The operator  $\cup$  is used to avoid multiple scheduling for the common groups in  $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ . Due to the presence of users' deadlines, unlike P1, two intuitive questions arise in P2. That is, in order to squeeze each user's transmission time as much as possible, should the appearance of the groups containing the same user be consecutive, namely consecutive group scheduling for every user? Should the user with the most stringent deadline be served first? Next, we give a negative answer for the first question by a counterexample, and use Lemma 2 to provide a positive answer for the second question.

Firstly, We conclude that consecutive group scheduling can be valid for some users but may not always hold for every user. Consecutive scheduling for a user means that all the activated groups for serving a user must be scheduled consecutively in the scheduling order, e.g., like users 1, 2, and 3 in Schedule II in Fig. 1, otherwise it is a nonconsecutive schedule, like user 1 in Schedule I in Fig. 1. Although Fig. 1 suggests that a consecutive schedule is better than a nonconsecutive one, in some instances the consecutive schedule may fail to guarantee the feasibility as well as the optimality. An example is shown below.

*Example 1:* Consider a nonconsecutive schedule for user  $k$  in a specific order  $\dots, \bar{u}_k, u', \dots, u'', \hat{u}_k, \dots$ , where  $\bar{u}_k$  and  $\hat{u}_k$  are the only two scheduled groups for user  $k$ , and groups  $u'$  and  $u''$  contain the other users. Suppose all the users' deadlines are satisfied by using these groups and this order. If we simply exchange the position between  $u'$  and  $\hat{u}_k$ , or between  $\bar{u}_k$  and  $u''$ , to form a consecutive schedule for user  $k$ , either of the changes may make some users' transmission time longer due to postponing groups  $u'$  or  $\bar{u}$  in the sequence and thus may cause deadline violation.

Regarding the second question, we conclude that it is an optimal principle for determining the scheduling order. We prove the following lemma, and derive an algorithm to determine the optimal scheduling order for P2. In general, changing the order for a user not only affects the user itself but also may influence whether other users' deadlines can be met. This is one of the difficulties in determining the optimal scheduling order.

---

**Algorithm 1** Determine Optimal Scheduling Order for Any Feasible Schedule in P2

---

**Input:** Deadlines  $T_1, \dots, T_K$ , a feasible schedule with groups in set  $\mathcal{G}^*$

**Output:** Optimal order  $\mathbf{o}^*$

---

- 1:  $\mathbf{o}^* = \emptyset$
  - 2: Sort  $T_1, \dots, T_K$  in an ascending order (suppose  $T_1 \leq \dots \leq T_K$  here for simplicity).
  - 3: **for**  $k = 1, \dots, K$  **do**
  - 4:    $\mathbf{o}^* = \mathbf{o}^* \cup \{\mathcal{G}_k \cap \mathcal{G}^*\}$
  - 5:    $\mathcal{G}^* = \mathcal{G}^* \setminus \{\mathcal{G}_k \cap \mathcal{G}^*\}$
  - 6:   **if**  $\mathcal{G}^* = \emptyset$  **then**
  - 7:     break.
  - 8: **Return:**  $\mathbf{o}^*$
- 

*Lemma 1:* Defining the strictest deadline among the users of a group as the group deadline, i.e.,  $\min_{k \in \mathcal{K}_u} T_k$ , then there exists an optimal scheduling where the active groups are organized following the ascending order of the group deadlines.

*Proof:* Suppose in the optimum, two groups,  $u'$  with the group deadline  $T_1$  and  $u''$  with its group deadline  $T_2$ , have  $T_1 < T_2$  but  $u'$  is scheduled after  $u''$ , i.e., schedule 1:  $\dots, u'', \dots, u', \dots$ , which violates the defined ascending order. Note that due to the optimality in schedule 1, the strictest deadline  $T_1$  has not expired by the end of the activation time of group  $u'$ . Based on schedule 1, we then construct a new schedule by moving group  $u''$  to be right after  $u'$ , i.e., schedule 2:  $\dots, u', u'', \dots$ , and correspondingly, group  $u'$ , as well as the groups between of  $u''$  and  $u'$  in schedule 1, are shifted earlier in the sequence to fill the gap of  $u''$ . The rest of the groups after  $u'$  in schedule 1 remain unchanged. Since  $T_1 < T_2$ , group  $u''$  still meets its group deadline  $T_2$  in schedule 2 which is feasible and also optimal as the consumed energy remains. Repeating the process for the groups that are not organized as the order defined by the lemma, we obtain an optimal schedule with this group ordering, and the conclusion follows. ■

Lemma 1 concludes the principle for deciding an optimal scheduling order in P2. The optimal order in P2 may not necessarily be unique, but adopting the order defined in Lemma 1 can always lead us to an optimal and feasible solution (if the problem is feasible). For any feasible schedule, we summarize the procedure of determining the optimal order for the schedule in Algorithm 1 and conclude the algorithm's optimality in Corollary 1. In Algorithm 1, the scheduled groups in set  $\mathcal{G}^*$  are organized into up to  $K$  segments. Suppose  $T_1 \leq \dots \leq T_K$ , the user with the strictest deadline, i.e., user 1 in this case, is firstly taken care of. All the active groups  $u \in \mathcal{G}_1 \cap \mathcal{G}^*$  are put in the first segment and scheduled in the first iteration. The groups' scheduling order in a segment can be arbitrary, and the order is recorded in  $\mathbf{o}^*$ . Then the scheduled groups are excluded from set  $\mathcal{G}^*$ . The procedure repeats until  $\mathcal{G}^*$  is empty.

*Corollary 1:* Given the optimal groups of P2, the optimal scheduling order can be determined by Algorithm 1 with polynomial-time complexity.

Given the formulations for the two NESP problems in P1 and P2, we then characterize the connection between the optimal energy  $E_1^*$  of P1 and  $E_2^*$  of P2 in the following remark.

*Remark 2:* Assuming the same input parameters for P1 and P2, if  $T_{tot} = \max\{T_1, \dots, T_K\}$ , then  $E_1^* \leq E_2^*$ . The reason can be explained as follows. Consider the same parameters in P1 and P2, and assume  $T_1 \leq \dots \leq T_K$ . If  $T_{tot}$  is set to be the maximum of the deadlines, i.e.,  $\max\{T_1, \dots, T_K\} = T_K = T_{tot}$ , P1 is a relaxed version of P2, since the scheduling order can be arbitrary in P1 and only the maximum deadline  $T_{tot}$  is present in P1, whereas P2 submits to extra deadlines  $T_1, \dots, T_{K-1}$  and a specific scheduling order. The two problems are equivalent, i.e.,  $E_1^* = E_2^*$ , when deadlines  $T_1, \dots, T_{K-1}$  in P2 are redundant, e.g., setting  $T_{tot} = T_1 = \dots = T_K$ . If the deadlines  $T_1, \dots, T_{K-1}$  are binding, the solution space of P2 is part of P1, and the conclusion follows.

#### IV. SOLUTION DEVELOPMENT: LEARNING-ASSISTED OPTIMIZATION

In this section, we first characterize the complexity and the difficulty in solving P1 and P2, then we propose a learning-assisted framework to efficiently solve P1 and P2, and show the advantages.

##### A. Characterizations for Complexity and Difficulty

The two NESPs are linear programming (LP) problems which can be solved by standard optimization tools, e.g., the simplex algorithm [24]. However, the linearity here does not conclude the problem's tractability. In fact, with the input size  $2^K - 1$  in P1 and P2, the computational complexity and the computational time of NESP increase exponentially with  $K$ . We derive the hardness results in Lemma 1.

*Proposition 1:* P1 and P2 are NP-hard.

*Proof:* By constructing a graph  $G$  with  $K$  nodes and treating each node as a user, the NP-hardness proof of P1 can be referred to the cell clustering and scheduling (CCS) problem in [19] which is proved to be NP-hard by a polynomial-time reduction from the fractional chromatic number in graph  $G$ . It can be proved that any two users connected by an edge in graph  $G$  will not be in the same group at the optimum of P1 instance. Therefore all the scheduled (optimal) groups in P1 correspond to independent sets in graph  $G$ . Solving the P1 instance gives the answers for the recognition version of fractional chromatic number. For P2, based on the conclusion in Remark 2, we construct a polynomial-time reduction from any instance of P1 by setting  $T_1 = \dots = T_K = T_{tot}$ . By doing so,  $T_1 = \dots = T_{K-1}$  can be always satisfied if the user  $K$ 's deadline, which becomes the total transmission deadline  $T_{tot}$ , is met. Hence a special case of P2 is at least as hard as P1, then in general P2 remains hard with the presence of multiple deadlines. ■

Note that both P1 and P2 fall into the domain of scheduling problems. For minimizing the energy consumed in data transmission, if TDMA is feasible, it is the optimal scheduling scheme [19], [21], e.g., when users' demand is low

and can be satisfied within deadline(s) by TDMA. In this paper, we are more focused on realistic scenarios with heavier data traffic and stricter time limits in data transmission. For these cases, TDMA fails, and the optimal scheduling is required.

In practical systems, stringent execution-time is required in real-time scheduling, thus P1 or P2 is expected to be solved efficiently. For many-user cases, the conventional iterative approaches may reach their limits in supporting online real-time network optimization. Solving such difficult problems to satisfactory performance would require a much longer span of computing time. Thus, understanding the practical limitations of the conventional approaches, we are motivated to explore new avenues in solution development.

In order to provide a reasonable algorithmic solution, we first investigate and reveal the major difficulties in efficiently solving P1 and P2. The first aspect is the exponentially increased variables. To determine the optimal groups, when  $K$  is large the optimal algorithms have to go through and evaluate a huge number of variables, e.g., the iterative procedure of identifying the column(s) with "the most negative reduced cost" in the column generation algorithm, or the successive pivot operations in the simplex algorithm [24]. On the other hand, by the LP theory, in the optimum, the number of scheduled groups is no more than the number of constraints [20]. One can observe that the number of constraints in P1 and P2 is linear in  $K$ . As a matter of fact, the NESP's output vector  $[t_1, \dots, t_u, \dots, t_U]$  will be very sparse, since most of the elements/groups in the vector will be zero/inactive. At the global optimum, most of the groups will not be used at all. Therefore to accelerate the optimization process, an effective way is to confine the searching space and let the algorithm avoid exploring those unused groups. However, it is not immediately clear which groups are not optimal and should be excluded from the  $2^K - 1$  candidates. This introduces the second difficulty which is the implicit combinatorial aspect in P1 and P2. In fact, although the variables in P1 and P2 are continuous, integer decisions have to be made, e.g., determining whether a user's data should be transmitted in group(s) or entirely delivered by TDMA. If the information of this combinatorial part can be known prior to the optimization process, the computational efficiency will be largely improved.

##### B. The Proposed Algorithmic Solution

We propose a learning-assisted optimization framework to provide high-quality and time-efficient solution. The idea is to apply learning-based approaches to help the optimal algorithm to tackle the most difficult and time-consuming part in the optimization. We train a deep neural network (DNN) for P1 or for P2, and let them learn how the optimal solutions, in terms of the patterns of optimal user grouping, behave in the optimization process. A maturely trained DNN can be used to provide guidance to confine the searching space of P1 and P2. The training of DNN is carried out at the pre-optimization stage such that the computational efforts can be significantly reduced for real-time applications and with satisfactory performance.



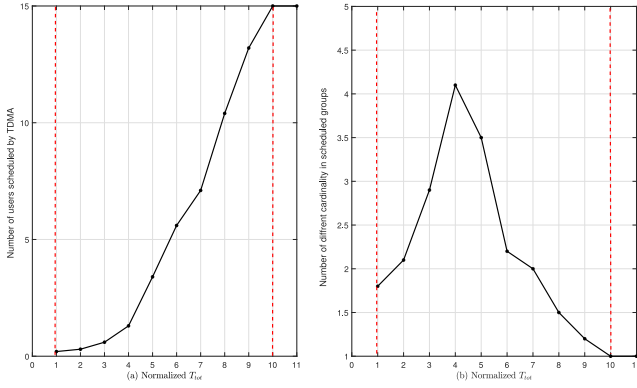


Fig. 2. An illustrative example: The features of the optimal groups.

The rationale behind this learning-based approach is that in many instances, the same types of optimization problems are usually solved repeatedly on a regular basis [11]. These problems share some common structure but differ in the input data. As a result, there exists relations between the input data and the optimization results. This provides an opportunity for machine-learning based approaches to learn this relationship, and then to predict the results, or to provide good initialization points for kick-starting heuristic-based optimization procedures. In the NESP problems, although the problems are of high complexity, one can still recognize the patterns between the input parameters and the scheduled groups in the optimum. We use a 15-user example in Fig. 2 to illustrate the features of the optimal groups in P1. From Fig. 2(a), when  $T_{tot}$  becomes large, TDMA transmission will be preferred to more users in the optimum. If the deadline is strict, almost all the users are scheduled in groups instead of TDMA since the optimal solution may have to choose the groups with large cardinality to guarantee the feasibility. Then the probability of using those small-cardinality groups can be very low. From Fig. 2(b), we observe that on average only very few types of cardinality are actually used in the optimum. Overall, around 1 to 4 cardinality types out of 15 are scheduled. The optimization process will be more efficient if the searching process can be more concentrated on the large-cardinality groups, instead of the whole solution space. In addition, towards high computational efficiency in real-time operations, the complexity of feed-forward operations of the DNN is mainly from the matrix multiplication, e.g., weight matrix times input matrix, the activation function, and the learning function [25], which are of low complexity in general, and can be executed fast in practice [26].

Considering the exponential number of groups in the optimization, it prohibits the DNN simply setting its output as the optimization variables  $t_1, \dots, t_U$  of P1 and P2, mainly due to the difficulties and the degraded performance in approximating the DNN's input-output relations with a huge number of variables [26]. Moreover, the solution feasibility of P1 and P2 will be sensitive to the prediction accuracy of the DNN. Any prediction error in DNN's output  $t_1, \dots, t_U$  can easily result in failures of satisfying all the constraints. Alternatively, we adopt an error-tolerable design in DNN predictions, such that

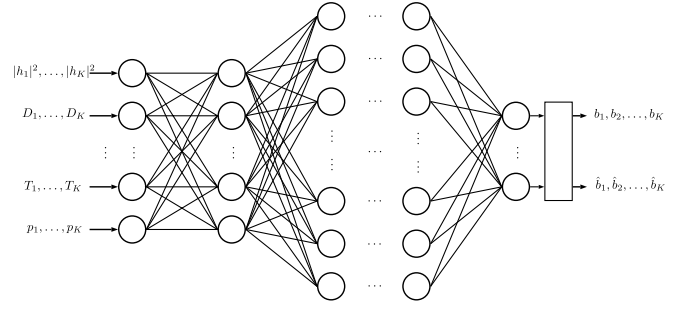


Fig. 3. Illustration of a fully connected DNN for P2.

the developed solution can benefit from the DNN prediction and meanwhile enable feasible solutions.

By our design, the DNN is expected to provide or learn two types of key information. One key information is formed by a  $K$ -dimension binary vector  $\mathbf{b} = [b_1, b_2, \dots, b_K]$ . The element  $b_k$  indicates whether user  $k$ 's demand  $D_k$  is transmitted alone in the whole scheduling period ( $b_k = 0$ ), i.e., TDMA, or delivered in any group(s) ( $b_k = 1$ ). The other information type is stored in another  $K$ -dimension binary vector  $\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_K]$ , where  $\hat{b}_k$  represents whether any of  $k$ -cardinality groups should be scheduled in the optimum ( $\hat{b}_k = 1$ ), or not used at all ( $\hat{b}_k = 0$ ). We summarize the input and output of the adopted DNN below, and an illustrative structure of the trained DNN is shown in Fig. 3.

**DNN Input:** Channel coefficients  $\mathbf{h} = [|h_1|^2, \dots, |h_K|^2]$ , user demand  $\mathbf{d} = [D_1, \dots, D_K]$ , user power  $\mathbf{p} = [p_1, \dots, p_K]$ , and transmission deadline  $T_{tot}$  specific for P1, and individual deadlines  $T_1, \dots, T_K$  specific for P2.

**DNN Output:** Prediction vectors  $\mathbf{b} = [b_1, b_2, \dots, b_K]$  and  $\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_K]$ .

The proposed learning-based optimization framework is summarized in Algorithm 2. In general, the optimization procedure consists of an offline DNN training phase and an online operation phase. In the offline training phase, taking P1 as an example, we use a tuple  $l_{P1}^s = (\mathbf{h}^s, \mathbf{d}^s, \mathbf{p}^s, T_{tot}^s, \mathbf{b}^s, \hat{\mathbf{b}}^s)$  to represent the  $s$ -th training sample for P1 ( $l_{P2}^s = (\mathbf{h}^s, \mathbf{d}^s, \mathbf{p}^s, \{T_1^s, \dots, T_K^s\}, \mathbf{b}^s, \hat{\mathbf{b}}^s)$  for P2). We generate realizations  $\mathbf{h}^s, \mathbf{d}^s, T_{tot}^s$ , and pre-optimized power  $\mathbf{p}^s$ . The information of  $\mathbf{b}^s$  and  $\hat{\mathbf{b}}^s$  is extracted from the optimal solution. We use a 4-user example to illustrate the procedure of preparing  $\mathbf{b}^s$  and  $\hat{\mathbf{b}}^s$  in the training set. As the first step we apply the exact algorithms, e.g., simplex or column generation algorithm, to obtain the optimal solution, say optimal groups  $\{1, 3\}, \{2, 3\}, \{1, 2\}, \{4\}$ . Next, the vectors  $\mathbf{b}^s$  and  $\hat{\mathbf{b}}^s$  will be formed as  $\mathbf{b}^s = [1, 1, 1, 0]$  and  $\hat{\mathbf{b}}^s = [1, 1, 0, 0]$  in the training set. The zero elements are due to the fact that user 4 is transmitted alone in TDMA, and all the 3-cardinality and 4-cardinality clusters are not scheduled in the optimal solution. By repeating the above data-generating process, we obtain the entire training set  $(l_{P1}^1, \dots, l_{P1}^S, l_{P2}^1, \dots, l_{P2}^S)$  for P1, where  $S$  is the maximum number of generated data samples in DNN training. Feeding the DNN by a sufficient large training set, the DNN is able to progressively optimize its weight values and learn the relation between the input and the output [26]. Analogous to the training set generation, the validation and test

---

**Algorithm 2** Learning-Assisted Optimization Framework for P1 and P2

---

**Input:**  $T_{tot}$  for P1,  $T_1 \dots T_K$  for P2, and  $D_1 \dots D_K$ ,  $|h_1|^2 \dots |h_K|^2$ ,  $p_1 \dots p_K$  for both P1 and P2.

**Output:**  $t_1^* \dots t_U^*$

---

- 1: **Offline Training Phase:**
  - 2: Generate training, validation, and test sets for P1 or P2.
  - 3: Complete the training of DNN1 for P1 or DNN2 for P2.
  - 4: **Online Operation Phase:**
  - 5: Take an input from the test set for P1 or P2.
  - 6: Obtain the predicted vectors  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  via DNN's output.
  - 7:  $M = \sum_{k=1}^K b_k / K$  and  $\hat{M} = \sum_{k=1}^K \hat{b}_k / K$
  - 8: **for**  $k = 1, \dots, K$  **do**
  - 9:   **if**  $b_k \notin \{0, 1\}$  and  $b_k > \alpha M$  (or  $\hat{b}_k > \alpha \hat{M}$ ) **then**
  - 10:      $b_k = 1$  (or  $\hat{b}_k = 1$ )
  - 11:   **else**
  - 12:      $b_k = 0$  (or  $\hat{b}_k = 0$ )
  - 13:  $\mathcal{U}^* = \mathcal{U}$
  - 14: **for**  $k = 1, \dots, K$  **do**
  - 15:   **if**  $b_k = 0$  **then**
  - 16:      $\mathcal{U}^* = \mathcal{U}^* \setminus \{u \in \mathcal{U}^* : k \in \mathcal{K}_u, |\mathcal{K}_u| > 1\}$
  - 17:   **if**  $\hat{b}_k = 0$  **then**
  - 18:      $\mathcal{U}^* = \mathcal{U}^* \setminus \{u \in \mathcal{U}^* : |\mathcal{K}_u| = k\}$
  - 19: Replace  $\mathcal{U}$  by  $\mathcal{U}^*$  in P1 or P2
  - 20: Obtain  $t_1^*, \dots, t_U^*$  by applying optimal algorithms to solve the restricted P1 or P2 with  $\mathcal{U}^*$ .
- 

sets are generated independently. For cross-validation process, the validation set is used to avoid over-fitting effect [25]. The test set (which is unseen during training) is then used in the after-training phase, for evaluating the performance of the DNN.

In the operation phase, taking an input from the test set and giving it to the well-trained DNN, the DNN can efficiently provide the required key information, i.e., the output vectors  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ . Note that there may exist fractional elements in  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ . We then design a rounding approach to convert those fractional elements to binary. We firstly compute the mean values  $M$  and  $\hat{M}$  for  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ , respectively. If any fractional  $b_k > \alpha M$  or  $\hat{b}_k > \alpha \hat{M}$ , we set  $b_k = 1$  or  $\hat{b}_k = 1$ , respectively, otherwise zero, where  $\alpha > 0$  is a control parameter to balance the computational efficiency and the optimality. Relying on the binary  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  from the DNN's output, a considerably large amount of groups are expected to be excluded from  $\mathcal{U}$ , forming a small-scale candidates set  $\mathcal{U}^*$  for P1 or P2. By doing so, the proposed method is able to break the trend of exponentially increased complexity in P1 and P2. Then solving a small-scale P1 or P2 can be much more efficient than a large-scale one.

We remark that as long as the well-trained DNN can provide precise predictions or the predictions achieving the same accuracy effect, the proposed solution in Algorithm 2 guarantees global optimality. Analogous to  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  in DNN, let  $\mathbf{b}^*$  and  $\hat{\mathbf{b}}^*$  be the binary vectors derived from the optimal solution. When we say a predicted element  $b_k \in \mathbf{b}$  (or  $\hat{b}_k \in \hat{\mathbf{b}}$ ) is accurate, it does not necessarily mean that the value of  $b_k \in \mathbf{b}$

must be exactly same as  $b_k^*$  in optimal  $\mathbf{b}^*$ . We characterize an sufficient optimality condition in Lemma 2 below, and define  $b_k$  or  $\hat{b}_k$  is accurate if  $b_k - b_k^* \geq 0$  or  $\hat{b}_k - \hat{b}_k^* \geq 0$ , respectively.

**Lemma 2:** Algorithm 2 guarantees global optimality if  $\mathbf{b} - \mathbf{b}^* \succeq \mathbf{0}$  and  $\hat{\mathbf{b}} - \hat{\mathbf{b}}^* \succeq \mathbf{0}$ .

*Proof:* The rationale is that if the predicted vectors  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  can keep all the optimal groups in the restricted set  $\mathcal{U}^*$ , Algorithm 2's optimality will be guaranteed, and the predicted  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  are considered achieving the same effect of the accurate prediction. Specifically, if an element  $b_k^* = 0$  in the optimum, user  $k$  will be transmitted by TDMA and only one group  $u$  with  $\mathcal{K}_u = \{k\}$  is needed for user  $k$ . Then the predicted  $b_k$  can be either zero or one without affecting the optimality since in either case, group  $u$  will be in  $\mathcal{U}^*$ . This is because ideally, if the DNN's prediction is accurate, the DNN's output (after rounding) should be  $b_k = 0$ , e.g., firstly read  $b_k = 0.1$  from the DNN's output then round it to  $b_k = 0$  in Algorithm 2. In this case, the TDMA group  $\{k\}$  will be the only group for user  $k$  in  $\mathcal{U}^*$ . However, the DNN's predictions may not be completely accurate, and the rounding process may not guarantee to always round  $b_k$  to zero. For example, we may read  $b_k = 0.9$  from the DNN's output due to its imperfect predictions. Then the rounding process may round it to  $b_k = 1$  which mismatches with optimal  $b_k^* = 0$ . As a result, the proposed algorithm will keep all the groups containing user  $k$ , including the TDMA group  $\{k\}$ , in  $\mathcal{U}^*$ . Thus we have  $b_k > b_k^*$  or  $b_k = b_k^*$ .

If  $b_k^*$  is one, the accurate predicted value should be  $b_k = 1$ , otherwise some of the optimal groups could be removed from  $\mathcal{U}^*$  in Line 16, and these groups will be no longer considered in solving the optimization problem in Line 20, thus the algorithm loses the optimality. For  $\hat{b}_k^* \in \hat{\mathbf{b}}^*$ , the logic is analogous, hence the conclusion. ■

From Lemma 2, as long as the optimality condition holds, Algorithm 2 in fact equivalently transforms a large-scale optimization task to a small-scale one by precisely excluding non-optimal groups without loss of any optimality. In case of the condition violation in  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ , the designed control parameter  $\alpha$  can be scaled down to tolerate prediction errors and improve the prediction accuracy. For instance, suppose  $M = 0.5$ ,  $\alpha = 1$ , and accurate  $b_k^*$  should be one in the optimum, however, due to imperfect estimation we read  $b_k = 0.49$  from the DNN's output. In Algorithm 2,  $b_k$  will be rounded to zero as  $b_k < \alpha M$ , meaning that any groups involved by user  $k$  will be excluded from the optimization process for solving P1 or P2, though at least one group among them is clearly optimal. For this case, we can scale down  $\alpha$ , say 0.9, then  $b_k$  will be set one as  $b_k = 0.49 > \alpha M = 0.45$ , and all the groups containing user  $k$  will be searched in the last step of Algorithm 2.

## V. PERFORMANCE EVALUATION

In this section, we provide numerical results to illustrate the performance gain of NOMA compared to the interference-channel (IC) based scheduling [19], [20] and TDMA. We show the effectiveness of the proposed learning-assisted approach, in terms of complexity and computational time reduction,



TABLE I  
SIMULATION PARAMETERS

Parameter	Value
Carrier frequency	2 GHz
Channel bandwidth	1 MHz
Number of users	15-25
Path loss	COST-231-HATA
Shadowing (Log-normal)	8 dB standard deviation
Fading	Rayleigh fading
Noise power spectral density	-173 dBm/Hz
BS maximum transmit power	5 W
Power allocation for users	FTPC [22]

DNN's prediction accuracy, and the performance gap between Algorithm 2 and the optimum.

#### A. Experimental Setup

For DNN training, the adopted DNN consists of a  $N$ -node input layer, a  $N$ -node encoding layer for data regularization, two hidden layers with 100 and 50 neurons for further learning the relations between the input and the output, and a  $2K$ -node output layer, where  $N = 3K + 1$  for P1 and  $N = 4K$  for P2. The resilient back-propagation (RProp) algorithm is used as the learning heuristic in DNN training and the online operation phase [26]. Fig. 3 shows an illustrative structure of DNN2 for P2, and DNN1 for P1 is analogous (replacing  $T_1, \dots, T_K$  by  $T_{tot}$  in the input nodes). The simulation parameters are summarized in Table I. To prepare the power-allocation vectors  $\mathbf{p}$  in training sets, we adopt a low-complexity scheme “fractional transmit power control (FTPC)” [22] for power allocation among the users. In FTPC implementation, we scale the control parameter  $\alpha$ , ranging from zero to one, to optimize the user power  $\mathbf{p} = [p_1, \dots, p_K]$ , where  $\alpha = 0$  enables equal power allocation among  $p_1, \dots, p_K$ , and increasing  $\alpha$  results in more power to the users with inferior channel condition for fairness consideration. We remark that any other power allocation algorithm can also be adopted in training data generation.

#### B. Performance of NOMA in Energy Saving and Meeting Deadlines

To evaluate the performance of applying NOMA in reducing energy and in meeting deadlines, six scheduling schemes, i.e., NOMA for P1 and P2, TDMA for P1 and P2, and IC for P1 and P2, have been implemented for comparisons. Their optimal solutions are enabled by applying standard LP algorithms. In IC scheduling, e.g., [19] and [20], no SIC is performed in each group, thus the interference part in equation (3) is replaced by the sum of intra-group interference, i.e.,  $\sum_{j \in \mathcal{K}_u \setminus \{k\}} p_j |h_k|^2$ . The results of optimal energy are illustrated in Figs. 4 and 5 for P1 and P2, respectively, where uniform users' demand of 10 Mbits and 15-user cases are adopted. In both figures, we generate one thousand instances and show the average performance over the normalized time lines.

In Fig. 4, we compare the energy consumption with and without circuit energy for NOMA and IC schemes. The circuit energy is modeled as  $P_c \sum_{u \in \mathcal{U}} t_u$  where  $P_c$  is circuit

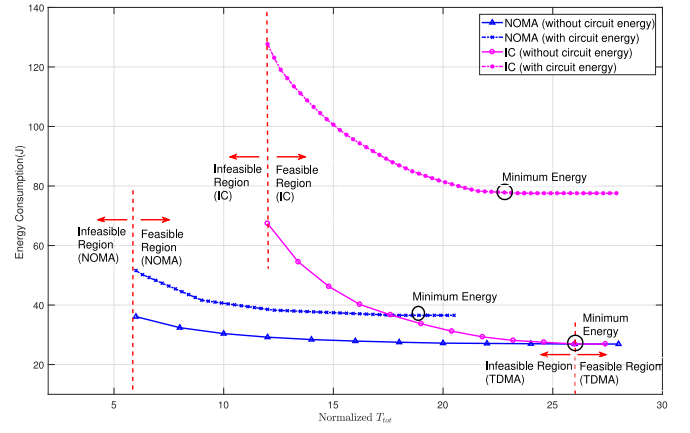


Fig. 4. Energy-saving performance in P1.

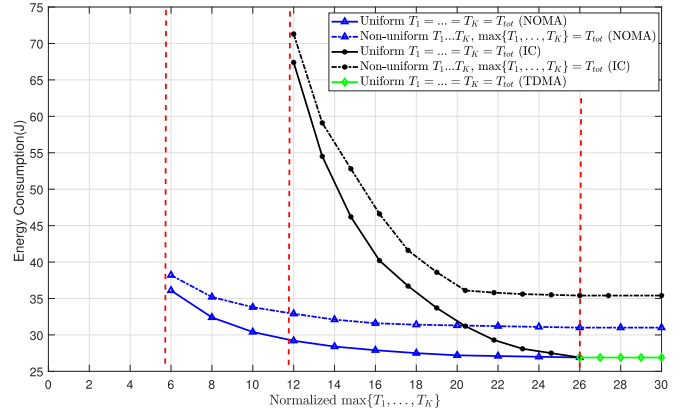


Fig. 5. Energy-saving performance in P2.

power and is set as 2 W in this simulation. We consider the circuit energy in Fig. 4 is linear with the transmission duration  $\sum_{u \in \mathcal{U}} t_u$ , and is independent of the transmission rate [27]. Then the objective function in P1 becomes  $\min_{t_u} \sum_{u \in \mathcal{U}} t_u \sum_{k \in \mathcal{K}_u} p_k + P_c \sum_{u \in \mathcal{U}} t_u$ . Firstly, for the two schemes without circuit energy (two solid lines), the range of  $T_{tot}$  is between interval  $[T_{feas}, T_{tdma}]$ , where  $T_{feas}$  is the minimum duration to keep the problem feasible, and  $T_{tdma}$  is the maximum transmission duration to keep the constraint  $\sum_{u \in \mathcal{U}} t_u \leq T_{tot}$  active. If  $T_{tot} < T_{feas}$  the problem becomes infeasible, and if  $T_{tot} \geq T_{tdma}$  TDMA will be optimal for P1. Within the interval  $T_{tot} \in [T_{feas}, T_{tdma}]$ , the TDMA scheme cannot deliver all the demand within deadline  $T_{tot}$ , and therefore user groups will be scheduled to save transmission time by allowing simultaneously serving multiple users. Compared to IC scheduling, the performance gain of adopting NOMA is significant. In general, the transmission energy in both NOMA (refer to the objective function in P1) and IC based scheduling will dramatically increase if more stringent deadline is imposed. However, the growth rate and the energy consumption in NOMA are much lower than IC. In addition, since part of the co-channel interference can be removed in each scheduled group, NOMA is able to achieve higher data rate and thus support the strict-deadline transmission. For instance, the normalized minimum  $T_{tot}$  in NOMA is 6, whereas IC becomes infeasible at  $T_{tot} = 12$ . Secondly, for the two schemes with

circuit energy (two dashed lines), the performance curves of NOMA and IC keep consistence with the two schemes without circuit energy, though the energy consumption is higher than the two solid lines. The energy consumption monotonically decreases when time  $T_{tot}$  increases. The minimum energy will not be able to further reduce if  $T_{tot}$  becomes redundant. Unlike the two solid-line schemes (without circuit energy), we remark that the optimal scheduling may not be TDMA when the minimum energy is achieved in the two dashed-line schemes. This is because with circuit energy, TDMA needs longer transmission time, thus it can result in higher circuit energy consumption. At the minimum-energy point, the transmission energy may not be minimum but the consumed circuit energy can be less by adopting non-TDMA schemes. Then the overall energy-saving performance may benefit.

Next, we examine the performance gain of NOMA in P2, and illustrate the relation between P1 and P2. The averaged results are shown in Fig. 5. As we expected, same as in P1, applying NOMA in multi-deadline scheduling can bring considerable performance benefits, compared to IC. In addition, as a verification of Remark 2, if we set uniform  $T_1 = \dots = T_K = T_{tot}$ , the energy of NOMA and IC in P1 is the same as their performance in P2. If we keep the longest deadline  $\max\{T_1, \dots, T_K\}$  as  $T_{tot}$ , say  $T_K = T_{tot}$ , and consider non-uniform deadlines, i.e., generate  $T_1, \dots, T_{K-1}$  by random values but bounded by  $T_{tot}$ , the consumed energy of NOMA and IC in P2 is bounded by P1, which is consistent with the conclusion in Remark 2. One may observe that when the longest deadline  $\max\{T_1, \dots, T_K\}$ , say  $T_K$ , becomes binding (see the left end of the curves), the gaps between the uniform and non-uniform cases are marginal. The reason is that if the maximum deadline becomes tight, then the room for scaling other users' deadlines becomes small, that is,  $T_1, \dots, T_{K-1}$  may need to be close to  $T_K$  otherwise the problems can be infeasible, leading to the small gaps with the cases of uniform  $T_1 = \dots = T_K = T_{tot}$ .

### C. Performance of Algorithm 2 in Computational Efficiency and Complexity Reduction

To illustrate the computational efficiency of the proposed Algorithm 2 in its online operation phase, we compare the CPU time (in seconds) in computations between Algorithm 2 (from Lines 5 to 20) and two optimal iterative algorithms, i.e., the simplex algorithm and the column generation algorithm. The former is a conventional algorithm for optimally solving LP, and the latter is proposed to improve the former's computational efficiency with guaranteed optimality. To provide a fair comparison, we implement and evaluate the three algorithms in a unified platform, MATLAB. All the three algorithms are applied to solve P1 and P2, and the averaged computational time per instance are shown in Table II.

In the results, from 5-user to 15-user cases, all the three algorithms can solve P1 and P2 very efficiently. For the cases of  $K > 15$ , the CPU time in Algorithm 2 keeps at the same magnitude as before, whereas the time in the other two algorithms exponentially increases with the number of users. The two conventional iterative algorithms may not be applicable in

TABLE II  
COMPARISON IN COMPUTATIONAL TIME

Cases	Simplex	Column Generation	Algorithm 2
$K = 5$	0.015	0.015	0.013
$K = 10$	0.078	0.043	0.019
$K = 15$	0.569	0.421	0.049
$K = 20$	156.1	113.9	0.075
$K = 21$	357.9	176.1	0.108
$K = 22$	2132	1175	0.194
$K = 25$	>3600	>3600	0.398

online optimization, though the column generation has largely reduced the computational time compared to the simplex algorithm. As can be foreseen, the computational efficiency of Algorithm 2 is insensitive with the increase of the input size. Overall, Algorithm 2 is promising to provide the optimized solution in time and with good performance in scalability.

To reveal the insights behind Table II, we use Fig. 6 and Fig. 7 to explain the reason why the proposed Algorithm 2 enables high efficiency in solving P1 and P2. We consider  $K = 5, 15, 25$  in Fig. 6 and Fig. 7. In the simplex and the column generation algorithms, to optimally solve P1 and P2, a number of  $|\mathcal{U}|$  candidates have to be searched, that is  $|\mathcal{U}| = 2^5 - 1 = 31$  for 5-user cases,  $|\mathcal{U}| = 2^{15} - 1 = 32767$  for 15-user cases, and  $|\mathcal{U}| = 2^{25} - 1 = 33554431$  for 25-user cases. It is evident that for larger  $K$ , searching such a huge number of groups is too time consuming. In contrast, a majority of the candidate groups in Algorithm 2 have been excluded in  $|\mathcal{U}^*|$  by relying on the predicted information from the DNN's output vectors  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ . One can observe that the magnitude of  $|\mathcal{U}^*|$  for large  $K$  has been reduced to hundreds on average, compared to millions in  $|\mathcal{U}|$ . As a result with the small-size  $|\mathcal{U}^*|$ , the optimization problem in Line 20 of Algorithm 2 can be solved efficiently, which therefore leads to a fast completion of the optimization process in Algorithm 2, even for the large-size inputs.

*Remark 3:* The high computational efficiency of Algorithm 2 displayed in Table II, Fig. 6 and Fig. 7, does not mean that we reduce the total computational efforts or simplify the optimization process in solving P1 and P2, like the most of heuristic algorithms. By our design, to facilitate real-time optimization we shift the majority of computations from the online operation phase to the offline data generation and training phase, such that the computational complexity in the online phase is moderate. In the offline training phase, the algorithm's solution quality can be improved or scaled by the computational efforts, e.g., scaling training set size  $S$ , which is a prior stage to the online optimization, and is less time-sensitive than the online phase.

From Fig. 6 and Fig. 7, for few-user cases, e.g.,  $K = 5$ , the performance gain of Algorithm 2 is marginal. Thus the proposed approach is preferred to apply to solve the large- $K$  cases, evident by its effectiveness in computations. From the results, Algorithm 2 is particularly efficient if the deadlines are loose. When imposing strict deadlines to Fig. 6 and Fig. 7, the average number of considered candidates in  $\mathcal{U}^*$  is moderately increased. In addition, one may notice that for the same  $K$ , the average cardinality of  $|\mathcal{U}^*|$  in P2 is larger than that in P1,

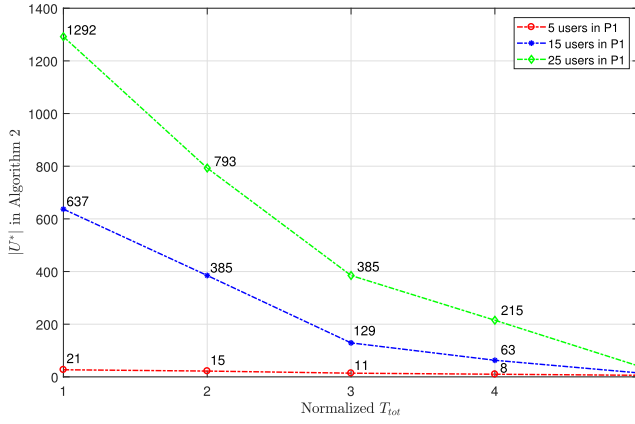


Fig. 6. Capabilities of Algorithm 2 in confining the number of candidates for P1. Benchmarks:  $2^5 - 1 = 31$  for 5-user cases,  $2^{15} - 1 = 32767$  for 15-user cases, and  $2^{25} - 1 = 33554431$  for 25-user cases.

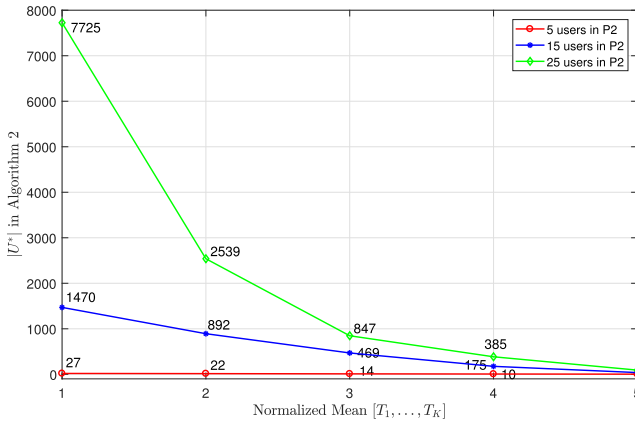


Fig. 7. Capabilities of Algorithm 2 in confining the number of candidates for P2. Benchmarks:  $2^5 - 1 = 31$  for 5-user cases,  $2^{15} - 1 = 32767$  for 15-user cases, and  $2^{25} - 1 = 33554431$  for 25-user cases.

though the CPU time for solving P1 and P2 remains at the same magnitude. The reason can be explained as follows. The cardinality of  $\mathcal{U}^*$  becomes smaller if more users are scheduled by TDMA, i.e., more zero elements in vector  $\mathbf{b}$ , and if the scheduled groups are more concentrated with fewer types of cardinality, i.e., more zero elements in vector  $\hat{\mathbf{b}}$ . In P1, this is the preferred type of group selections, e.g., in the optimum, more users will be in TDMA for large  $T_{tot}$ , and only large-cardinality groups will be used for small  $T_{tot}$ . However, by imposing multiple deadlines  $T_1, \dots, T_K$  in P2, the choices of the scheduled groups can be more diverse than in P1. For instance, some users in P2 would be scheduled by TDMA if their deadlines are loose, and some other users have to be scheduled in multiple groups with different cardinality due to their diverse deadlines. As a result, overall, more candidates may need to be searched for P2 than P1 in Algorithm 2.

#### D. Performance of Algorithm 2 in Approximating Optimum

Next, we evaluate the DNN's prediction accuracy which is an important aspect in determining the performance of Algorithm 2. In each instance, based on the optimal solution we derive the corresponding optimal  $\mathbf{b}^*$  and  $\hat{\mathbf{b}}^*$ , and compare

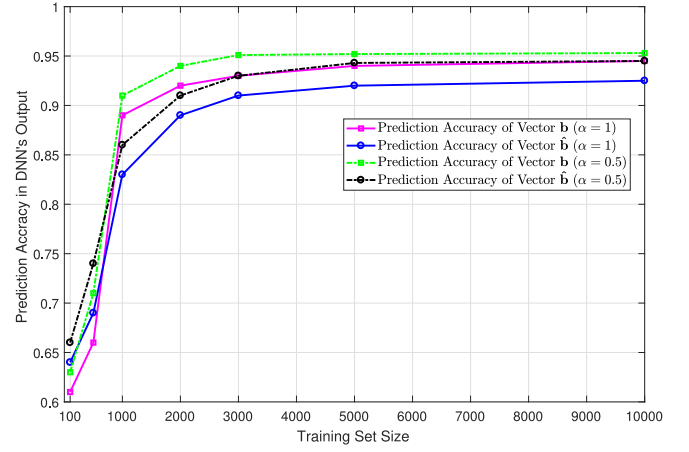


Fig. 8. Average prediction accuracy in DNN's output.

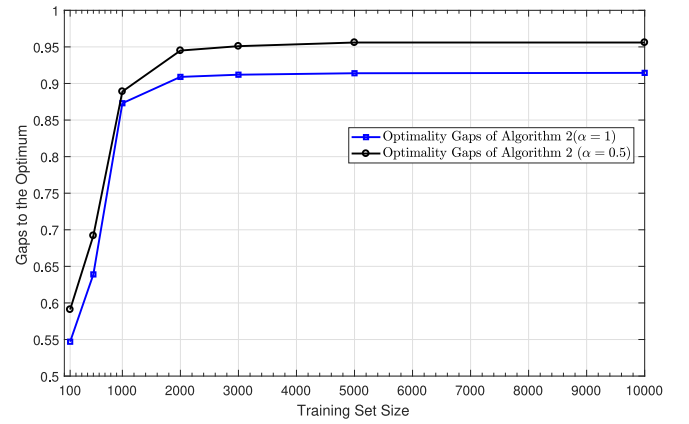


Fig. 9. Average gaps between the optimum and Algorithm 2.

them with the DNN's predicted vectors  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ . Then we show the average accuracy results in Fig. 8, for example, the value of "0.9" means that 90% of the elements in  $\mathbf{b}$  or  $\hat{\mathbf{b}}$  do not lead to loss of any optimality. From the results of Fig. 8, the DNN can achieve high prediction accuracy only after training by a considerable number of instances. We observe that, a well-trained DNN can provide an overall accurate prediction in  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ , around 93% to 96%. Moreover, the quality of prediction in both  $\mathbf{b}$  and  $\hat{\mathbf{b}}$  can be further improved by trading computational efforts via decreasing  $\alpha$ . By our design, small  $\alpha$  means that if we are not certain about excluding a type of groups, then we keep them in  $|\mathcal{U}^*|$ , in order to avoid wrongly excluding those optimal groups due to imperfect prediction in DNN. On average, the accuracy of  $\mathbf{b}$  is higher than  $\hat{\mathbf{b}}$ . This is because in general, the patterns of determining TDMA or non-TDMA for a user is easier to be recognized than the patterns of determining which cardinality of the groups should be used.

Then in Fig. 9, we show the Algorithm 2's ability in approaching the optimum with respect to the training progress. We compare the energy obtained in Algorithm 2 with the optimal energy to derive the gaps. For instance, "1.0" in Fig. 9 means that on average the energy performance in Algorithm 2 is consistent with the optimum. Based on the completely/relatively accurate information of  $\mathbf{b}$  and  $\hat{\mathbf{b}}$ , most

of the optimal groups are identified to be included in the restricted set  $\mathcal{U}^*$ , thus Algorithm 2 can achieve close-to-optimum performance. In Fig. 9, by relying on a well-trained DNN, the energy gaps between Algorithm 2 and the optimum is less than 9%. By adopting small  $\alpha$ , Algorithm 2 approaches to the optimum, but may pay more computational time than the case of  $\alpha = 1$ .

## VI. CONCLUSION

We considered applying NOMA in minimum-energy scheduling with practical deadline constraints. Considering the dynamic duration of transmission, and the queuing delay in data services, we formulated two resource scheduling problems, aiming at efficiently emptying the BS's queued data or providing timely services to satisfy users' demand by consuming less transmission energy. To deal with the practical issues of the high complexity in resource optimization and the stringent execution-time requirements in real-time operations, we proposed a learning-assisted optimization framework to provide an algorithmic solution for solving a class of scheduling problems. Numerical results firstly show the benefits of NOMA in saving energy and reducing transmission time. The results also demonstrate the promising performance of the proposed method in improving computational efficiency and optimality approximation. The developed algorithm is able to efficiently provide optimal scheduling solutions if the predictions in the deep learning model are precise. For the general cases, the solution can be close to the global optimum, around 5% to 9% gaps on average. The computational time is insensitive to the input size.

## REFERENCES

- [1] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [2] "Visual networking index: Global mobile data traffic forecast update, 2016–2021," San Jose, CA, USA, Cisco, White Paper, Feb. 2017. [Online]. Available: <https://www.cisco.com/>
- [3] Z. Ding *et al.*, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2181–2195, Oct. 2017.
- [4] L. Lei, D. Yuan, C. K. Ho, and S. Sun, "Power and channel allocation for non-orthogonal multiple access in 5G systems: Tractability and computation," *IEEE Trans. Wireless Commun.*, vol. 15, no. 12, pp. 8580–8594, Dec. 2016.
- [5] H. Hacı, H. Zhu, and J. Wang, "Performance of non-orthogonal multiple access with a novel asynchronous interference cancellation technique," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1319–1335, Mar. 2017.
- [6] L. You *et al.*, "Resource optimization with load coupling in multi-cell NOMA," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4735–4749, Jul. 2018.
- [7] G. Song and X. Wang, "Comparison of interference cancellation schemes for non-orthogonal multiple access system," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, 2016, pp. 1–5.
- [8] L. Lei, D. Yuan, and P. Värbrand, "On power minimization for non-orthogonal multiple access (NOMA)," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2458–2461, Dec. 2016.
- [9] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.
- [10] S. Sesia, I. Toufik, and M. Baker, *LTE: The UMTS Long Term Evolution*. Chichester, U.K.: Wiley, 2011.
- [11] F. Calabrese, L. Wang, E. Ghadimi, G. Peters, and P. Soldati. (2017). *Learning Radio Resource Management in 5G Networks: Framework, Opportunities and Challenges*. [Online]. Available: <https://arxiv.org/pdf/1611.10253.pdf>
- [12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [13] H. Sun *et al.*, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–6.
- [14] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–7.
- [15] L. Lei *et al.*, "A deep learning approach for optimizing content delivering in cache-enabled HetNet," in *Proc. IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2017, pp. 449–453.
- [16] L. Lei, T. X. Vu, L. You, S. Fowler, and D. Yuan, "Efficient minimum-energy scheduling with machine-learning based predictions for multiuser MISO systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [17] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Interference constraint active learning with uncertain feedback for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4654–4668, Jul. 2017.
- [18] A. Pantelidou and A. Ephremides, "The scheduling problem in wireless networks," *J. Commun. Netw.*, vol. 11, no. 5, pp. 489–499, Oct. 2009.
- [19] L. Lei, D. Yuan, C. K. Ho, and S. Sun, "Optimal cell clustering and activation for energy saving in load-coupled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6150–6163, Nov. 2015.
- [20] V. Angelakis, A. Ephremides, Q. He, and D. Yuan, "Minimum-time link scheduling for emptying wireless systems: Solution characterization and algorithmic framework," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1083–1100, Feb. 2014.
- [21] Q. He, D. Yuan, and A. Ephremides, "On optimal link scheduling with deadlines for emptying a wireless network," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 461–465.
- [22] Y. Saito, A. Benjebbour, Y. Kishiyama, and T. Nakamura, "System-level performance evaluation of downlink non-orthogonal multiple access (NOMA)," in *Proc. IEEE PIMRC*, Sep. 2013, pp. 611–615.
- [23] M. Shirvanimoghaddam, M. Condoluci, M. Dohler, and S. J. Johnson, "On the fundamental limits of random non-orthogonal multiple access in cellular massive IoT," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2238–2252, Oct. 2017.
- [24] K. Murty, *Linear Programming*. New York, NY, USA: Wiley, 1983.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [26] V. Sze, Y. Chen, T. Yang, and J. Emer. (Mar. 2017). *Efficient Processing of Deep Neural Networks: A Tutorial and Survey*. [Online]. Available: <http://arxiv.org/abs/1703.09039>
- [27] S. Luo, R. Zhang, and T. J. Lim, "Joint transmitter and receiver energy minimization in multiuser OFDM systems," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3504–3516, Oct. 2014.



**Lei Lei** (S'12–M'17) received the B.Eng. and M.Eng. degrees from Northwestern Polytechnic University, Xi'an, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Department of Science and Technology, Linköping University, Sweden, in 2016. Since 2016, he has been a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. He was a Research Assistant with the Institute for Infocomm Research, A\*STAR, Singapore, in 2013, for six months. His current research interests include resource allocation and optimization in 5G-satellite networks, wireless caching, energy-efficient communications, and machine learning in wireless communications. He was a recipient of the IEEE Sweden Vehicular Technology-Communications-Information Theory Joint Chapter Best Student Journal Paper Award in 2014.





**Lei You** (S'15) received the B.Eng. and M.Eng. degrees from Qingdao University, China, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Technology, Uppsala University, Sweden. He was a Visiting Researcher with Ranplan Wireless Network Design, Ltd., U.K., from 2015 to 2016, and Converge ICT, Athens, in 2016. He served as a Project Manager of the EU Horizon 2020 Marie-Sklodowska Curie Project DECADE from 2015 to 2016. His current research interests include math-

ematical optimization and machine learning in the domain of mobile communications and networking.



**Qing He** (S'11–M'16) received the B.Sc. and M.Sc. degrees in electrical engineering from Nanjing University, China, in 2001 and 2004, respectively, and the Ph.D. degree from the Department of Science and Technology, Linköping University, Sweden, in 2016. She is currently a Post-Doctoral Researcher with the Department of Network and Systems Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden. She has a second degree in Finance and had been a Software

Designer and a System Engineer with Lucent Technologies Bell Labs until 2011. Her current research interests include wireless network optimization and information theory.



**Thang X. Vu** (S'11–M'15) was born in Hai Duong, Vietnam. He received the B.S. and M.Sc. degrees in electronics and telecommunications engineering from the VNU University of Engineering and Technology, Vietnam, in 2007 and 2009, respectively, and the Ph.D. degree in electrical engineering from the University of Paris-Sud, France, in 2014. From 2007 to 2009, he was with the Department of Electronics and Telecommunications, VNU University of Engineering and Technology as a Research Assistant. From 2010 to 2014, he

was with the Laboratory of Signals and Systems, a joint Laboratory of CNRS, CentraleSupélec, and the University of Paris-Sud XI, France. From 2014 to 2016, he was a Post-Doctoral Researcher with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore. He is currently a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. He received the Allocation de Recherche Fellowship to study Ph.D. degree in France in 2010. His research interests are in the field of wireless communications, with particular interests of cache-assisted 5G, machine learning for communications, cloud radio access networks, and resources allocation and optimization.



**Symeon Chatzinotas** (S'06–M'09–SM'13) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003 and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Surrey, U.K., in 2006 and 2009, respectively. He is currently the Deputy Head of the SIGCOM Research Group, Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg, Luxembourg, and a Visiting Professor with the University of Parma, Italy. He was involved

in numerous research and development projects for the Institute of Informatics Telecommunications, National Center for Scientific Research Demokritos, the Institute of Telematics and Informatics, Center of Research and Technology Hellas, and the Mobile Communications Research Group, Center of Communication Systems Research, University of Surrey. He has over 300 publications, 2500 citations, and an *H*-index of 27 according to Google Scholar. His research interests include multiuser information theory, co-operative/cognitive communications, and wireless networks optimization. He was a co-recipient of the 2014 Distinguished Contributions to Satellite Communications Award, and the Satellite and Space Communications Technical Committee, the IEEE Communications Society, and the CROWCOM 2015 Best Paper Award.



**Di Yuan** (M'03–SM'15) received the M.Sc. degree in computer science and engineering and the Ph.D. degree in optimization from the Linköping Institute of Technology in 1996 and 2001, respectively. After his Ph.D., he has been an Associate Professor and then a Full Professor with the Department of Science and Technology, Linköping University, Sweden. In 2016, he joined Uppsala University, Sweden, as a Chair Professor. His current research mainly addresses network optimization of 4G and 5G systems, and capacity optimization of wireless

networks. He was a Guest Professor with the Technical University of Milan, Politecnico di Milano, Italy, in 2008 and a Senior Visiting Scientist with Ranplan Wireless Network Design Ltd., U.K., in 2009 and 2012. In 2011 and 2013, he has been part time with Ericsson Research, Sweden. In 2014 and 2015, he was a Visiting Professor with the University of Maryland, College Park, MD, USA. He is an Area Editor of *Computer Networks*. He has been in the management committee of four European Cooperation in field of Scientific and Technical Research actions, Invited Lecturer of European Network of Excellence EuroNF, and a Principal Investigator of several European FP7 and Horizon 2020 projects. He was a co-recipient of the IEEE ICC'12 Best Paper Award and the Supervisor of the Best Student Journal Paper Award by the IEEE Sweden Joint VT-COM-IT Chapter in 2014.



**Björn Ottersten** (S'87–M'89–SM'99–F'04) was born in Stockholm, Sweden, in 1961. He received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986 and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1990. He has held research positions with the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, the Katholieke Universiteit Leuven, Leuven, Belgium, and the University of

Luxembourg, Luxembourg. From 1996 to 1997, he was the Director of Research with ArrayComm, Inc., a start-up in San Jose, CA, USA, based on his patented technology. In 1991, he was appointed a Professor of signal processing with the KTH Royal Institute of Technology, Stockholm, Sweden, where he was the Head of the Department for Signals, Sensors, and Systems from 1992 to 2004 and the Dean of the School of Electrical Engineering from 2004 to 2008. He is currently the Director of the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. As Digital Champion of Luxembourg, he acts as an Adviser to the European Commission. He was a recipient of the IEEE Signal Processing Society Technical Achievement Award in 2011, the European Research Council advanced research grant twice, in 2009–2013 and in 2017–2022, the IEEE Signal Processing Society Best Paper Award in 1993, 2001, 2006, and 2013 for his co-authored journal papers, and seven other IEEE conference papers best paper awards. He has served as the Editor-in-Chief for the *EURASIP Signal Processing Journal*, an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the Editorial Board of the *IEEE Signal Processing Magazine*. He is currently an Editorial Board Member of the *EURASIP Journal of Advances Signal Processing* and the *Foundations and Trends in Signal Processing*. He is a fellow of EURASIP.