

Joint User Grouping and Power Allocation for MISO Systems: Learning to Schedule

Yaxiong Yuan, Thang X. Vu, Lei Lei, Symeon Chatzinotas, and Björn Ottersten

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg
Emails: {yaxiong.yuan; thang.vu; lei.lei; symeon.chatzinotas; bjorn.ottersten@uni.lu }

Abstract—In this paper, we address a joint user scheduling and power allocation problem from a machine-learning perspective in order to efficiently minimize data delivery time for multiple-input single-output (MISO) systems. The joint optimization problem is formulated as a mixed-integer and non-linear programming problem, such that the data requests can be delivered by minimum delay, and the power consumption can meet practical requirements. For solving the problem to global optimum, we provide a solution to decouple the scheduling and power optimization. Due to the problem’s inherent hardness, the optimal solution requires exponential complexity and time in computations. To enable an efficient and competitive solution, we propose a learning-based approach to reduce data delivery time and solution’s computational delay, where a deep neural network is trained to learn and decide how to optimize user scheduling. In numerical study, the developed optimal solution can be used for performance benchmarking and generating training data for the proposed learning approach. The results demonstrate the developed learning based approach is able to significantly improve the computation efficiency while achieves near optimality.

Index Terms—Time minimization, machine learning, power allocation, user scheduling.

I. INTRODUCTION

The upcoming 5G communication system is envisioned to support high-speed and low-latency data transmission for serving a massive number of connected devices. When users’ requests for data-hungry services arrive, how to satisfy their large-volume data demand within acceptable delay, and how to reduce the latency in data queuing and transmission become the important performance metrics in many application scenarios [1]. Developing intelligent transmission schemes and advanced resource scheduling algorithms are considered as a key enabler to achieve the strict performance requirements. On this track, there are numbers of studies in the literature that develop sophisticated scheduling algorithms. A related topic is transmission scheduling without power control. In [2], the authors investigated a minimum-length scheduling problem for generic wireless networks. The problem’s NP-hardness, optimality conditions, and a set of scheduling algorithms have been studied. The authors in [3] developed a column-generation based scheduling algorithm to complete all the data transmission within a transmission deadline.

For joint transmission scheduling with power control, the authors in [4] proposed centralized and distributed scheduling approaches to improve energy efficiency. In [5], the authors

studied joint optimization of link scheduling and power control for energy minimization. The optimality conditions for using time division multiple access (TDMA) transmission were derived. Compared to the scheduling problem without power control, the joint optimization for user scheduling and power control is even challenging. This is because the two components, i.e., finding the optimal user groups and the optimal power allocation among users, are mutually coupled in decision making, which typically leads to non-linear optimization [5] and thus results in difficulties to deliver optimal solutions [6].

In general, the scheduling optimization problems are hard to solve [2], [3]. Due to the problem’s inherent hardness, the high computational complexity and long computation time limit the algorithm’s applicability in practice. The execution-time for performing an algorithm in real-time systems is typically stringent, e.g., during a scheduling period, one should return the optimized results within milliseconds [1]. However, executing optimal algorithms may require considerable computational capabilities and time. It may not be practical to apply a sophisticated algorithm to real-time operations. An optimizer has to balance the algorithm’s time efficiency and the solution quality. In this paper, from a machine-learning perspective, we investigate an alternative trade-off solution for joint scheduling and power control. Machine/deep learning techniques have received much attention in wireless communications over the past few years [7], [8], [9], [10]. In [8], [9], [10], without considering power control the authors proposed a set of learning based optimization approaches to improve the scheduling performance and reduce the computational complexity.

In this paper, we tackle the joint scheduling and power control problem for multiple-input single-output (MISO) systems from a machine learning perspective. We explore the benefits of applying deep learning based approaches in minimum-length scheduling and power allocation. Firstly, we formulate the joint optimization task as an integer non-linear programming problem, where the optimization between user grouping and power allocation is mutually dependent. Secondly, we provide an approach to decouple these two parts and obtain optimal solutions for performance benchmarking. The optimization approach, however, imposes an exponential computing complexity, which may limit its capability in real-time applications. Thirdly, we design a learning-based algorithm to enable fast execution in decision making and in the mean-

while provide competitive performance. The derived optimal solutions are used to prepare data sets for training a deep neural network to learn the optimal solution in grouping and scheduling. Finally, we use numerical results to demonstrate the promising capabilities of the proposed learning approach in approximating optimal scheduling solution and reducing computational time.

The rest of the paper is organized as follows. Section II presents the system model for MISO and signal transmission. Section III formulates the joint user scheduling and power control problem with quality of service constraints (QoS). The method for obtaining global optimal solution is characterized. Section IV presents a learning based algorithm. Numerical results are demonstrated in Section V. Conclusions are given in Section VI. Throughout this paper, we use the following notations. Normal letters represent scalar quantities, bold face letters indicate vectors, and boldface uppercase letters designate matrices. The superscript $(\cdot)^H$ stands for Hermitian transpose and the two-norm of a vector is denoted by $\|\cdot\|$.

II. SYSTEM MODEL

We consider a downlink MISO system including an L -antenna base station (BS) and K single-antenna users denoted as $\mathcal{K} = \{1, \dots, k, \dots, K\}$. We consider all the users share a common communication channel in transmission. The k -th user requests a content of length Q_k bits. We assume the wireless channel follows the block Rayleigh fading.

A. User Scheduling

In order to mitigate co-channel interference and improve the transmission efficiency, the dynamic user-group scheduling and precoding strategy are adopted. Let g denote a user group and \mathcal{K}_g denotes the users included in group g . When group g is scheduled, the BS will deliver the requested contents to all the users in \mathcal{K}_g until all the transmissions for \mathcal{K}_g are complete. In total, there are $G = 2^K - 1$ possible candidate groups by enumerating all of the user groups, and the union of all the candidates is denoted by $\mathcal{G} = \{1, \dots, g, \dots, G\}$. For instance, when $K = 3$, we can list all the candidate groups $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$.

For satisfying all the users' demands, some groups will be selected from \mathcal{G} and scheduled in a sequential manner. The selected groups form a combination \mathcal{C} which covers all the users, and each user is scheduled once. It is noteworthy that $G = 2^K - 1$ becomes huge if K is large. In previous works, some simple schemes are proposed. One of them is time division multiple access (TDMA) which is a special case by setting $G = K$ and $|\mathcal{K}_g| = 1$. Another scheme is to put all users in one group, which means $G = 1$ and $|\mathcal{K}_g| = K$. Although all the users can be served simultaneously, the interference among the scheduled users could be severe. Both of the simple schemes have limitations in achieving desirable performance. In this work, we do not predefine the scheduling strategy but solve an optimization problem to determine the optimal group combination.

B. Transmission Model

The channel vector of user k is denoted by $\mathbf{h}_k \in \mathbb{C}^{L \times 1}$, where L is the number of antennas. We assume \mathbf{h}_k follows circular-symmetric complex Gaussian distribution $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_{h_k}^2 \mathbf{I}_L)$, where $\sigma_{h_k}^2$ is the parameter related to the path loss between the BS and user k . For the purpose of eliminating the mutual interference, the BS takes the precoding operation before broadcasting to users. Denote x_k^g as the modulated signal and $\mathbf{w}_k^g \in \mathbb{C}^{L \times 1}$ as the precoding vector for user k in group g . The received signal for user $k \in \mathcal{K}_g$ can be expressed as

$$y_k^g = \mathbf{h}_k^H \mathbf{w}_k^g x_k^g + \sum_{i \in \mathcal{K}_g \setminus \{k\}} \mathbf{h}_k^H \mathbf{w}_i^g x_i^g + n_k, \quad (1)$$

where the first and second terms in (1) represent the desired signal and the inter-user interference respectively. n_k is Gaussian noise with zero mean and variance σ^2 . The signal-to-interference-plus-noise ratio (SINR) for user $k \in \mathcal{K}_g$ is given by $\text{SINR}_k^g = \frac{|\mathbf{h}_k^H \mathbf{w}_k^g|^2}{\sum_{i \in \mathcal{K}_g \setminus \{k\}} |\mathbf{h}_k^H \mathbf{w}_i^g|^2 + \sigma^2}$. The achievable data rate can be expressed as

$$R_k^g = B \log_2 (1 + \text{SINR}_k^g), k \in \mathcal{K}_g. \quad (2)$$

With respect to the delivery time, once the scheduling combination \mathcal{C} is determined, the selected groups will be scheduled sequentially. Denote t_g as the transmission duration when group g is scheduled. The transmission for a scheduled group g will last until all the users' requests in \mathcal{K}_g are satisfied. t_g is given as

$$t_g = \max_{k \in \mathcal{K}_g} \frac{Q_k}{R_k^g}. \quad (3)$$

The total delivery time is defined as $T_{tot} = \sum_{g \in \mathcal{C}} t_g$.

III. JOINT SCHEDULING AND POWER ALLOCATION PROBLEM

We consider a joint user scheduling and power allocation problem in order to minimize the total delivery time. The zero-forcing (ZF) precoding is adopted for each group due to its low computational complexity. For each scheduled group g , denote \mathbf{H}_g as the channel coefficients from the BS's antennas to the users in g , which is a $|\mathcal{K}_g| \times L$ matrix. Under the ZF design, the beamforming vector for user k in group g is of the form $\mathbf{w}_k^g = \sqrt{p_k^g} \tilde{\mathbf{w}}_k^g$, where p_k^g is the power allocation for user k in group g and $\tilde{\mathbf{w}}_k^g$ is the k -th column of the ZF matrix $\mathbf{H}_g^H (\mathbf{H}_g \mathbf{H}_g^H)^{-1}$. It is worth noting that under the ZF design, the inter-user interference is canceled, i.e., $|\mathbf{h}_k^H \mathbf{w}_j^g| = \delta_{kj}$. Then the achievable rate for user k in group g is given as

$$R_k^g = B \log_2 (1 + p_k^g / \sigma^2), k \in \mathcal{K}_g. \quad (4)$$

The transmit power of user k is $p_k^g \beta_k^g$, where $\beta_k^g = \|\tilde{\mathbf{w}}_k^g\|^2$, $\forall k \in \mathcal{K}_g$.

A. Problem Formulation

The joint user scheduling and power allocation is formulated in P0. We introduce two sets of variables, $z_g \in \{0, 1\}$, $\forall g \in \mathcal{G}$ and p_k^g , $\forall k \in \mathcal{K}, \forall g \in \mathcal{G}$. The binary variable z_g is used to indicate if group g is scheduled ($z_g = 1$) or not ($z_g = 0$). The continuous variable $p_k^g > 0$ represents the power allocation among the users in group g .

$$\text{P0: Minimize}_{z_g \in \{0,1\}, p_k^g > 0} \sum_{g \in \mathcal{G}} z_g \max_{k \in \mathcal{K}_g} \frac{Q_k}{R_k^g} \quad (4a)$$

$$\text{s.t. } R_k^g \geq \eta_k, \forall k \in \mathcal{K}_g, \forall g \in \mathcal{G} \quad (4b)$$

$$\sum_{k \in \mathcal{K}_g} \beta_k^g p_k^g \leq P_{tot}, \forall g \in \mathcal{G} \quad (4c)$$

$$\sum_{g \in \mathcal{G}} a_{kg} z_g = 1, \forall k \in \mathcal{K} \quad (4d)$$

The objective in P0 is to minimize the total data transmission length, where the duration of scheduling group g depends on $\max_{k \in \mathcal{K}_g} \frac{Q_k}{R_k^g}$. Constraint (4b) is to guarantee users' minimum rate η_k in their transmission period. The second constraint (4c) restricts that the power consumption in each group cannot exceed power budget P_{tot} . In constraints (4d), we consider that each user is scheduled once to reduce the implementation complexity and the signaling overhead in practice, where binary parameters $a_{kg} = 1$ indicates group g containing user k , otherwise 0.

B. Optimal solution

In general, P0 is a mixed integer non-linear programming problem. To enable the optimal solution, we observe that the problem can be decomposed to two steps. In the first step, we enumerate all the groups and optimize the power allocation within each group such that the QoS constraints (4b) and (4c) are satisfied for each group and users. In the second step, we select a combination of groups, which leads to the minimum transmission time, and covers all the users. In addition, each user appears only once in the combination.

Firstly, for each group g , the intra-group power allocation problem can be formulated in P1(g):

$$\text{P1}(g) : \text{Minimize}_{p_k^g > 0} \max_{k \in \mathcal{K}_g} \frac{Q_k}{R_k^g} \quad (5)$$

$$\text{s.t. } R_k^g \geq \eta_k, \forall k \in \mathcal{K}_g \quad (5a)$$

$$\sum_{k \in \mathcal{K}_g} \beta_k^g p_k^g \leq P_{tot}, \quad (5b)$$

By introducing a variable T_g , problem P1(g) is equivalent to the following problem:

$$\text{P1}'(g) : \text{Minimize}_{p_k^g > 0, T_g \geq 0} T_g \quad (6)$$

$$\text{s.t. } T_g \geq \frac{Q_k}{\log_2(1 + p_k^g/\sigma^2)}, \forall k \in \mathcal{K}_g \quad (6a)$$

$$\log_2(1 + p_k^g/\sigma^2) \geq \eta_k, \forall k \in \mathcal{K}_g \quad (6b)$$

$$\sum_{k \in \mathcal{K}_g} \beta_k^g p_k^g \leq P_{tot}, \quad (6c)$$

The constraint (6a) is equivalent to,

$$\log_2(1 + p_k^g/\sigma^2) - \frac{Q_k}{T_g} \geq 0, \forall k \in \mathcal{K}_g, \quad (7)$$

which is a convex constraint. Thus P1'(g) is a convex problem, and can be solved by standard tools [11]. The next optimization task is to select a group combination leading to the minimum transmission time and covering all the users once. The problem can be formulated by an integer linear programming (ILP) problem in P2.

$$\text{P2: Minimize}_{z_g \in \{0,1\}} \sum_{g \in \mathcal{G}} z_g t_g \quad (8)$$

$$\text{s.t. } \sum_{g \in \mathcal{G}} a_{kg} z_g = 1, \forall k \in \mathcal{K} \quad (8a)$$

After solving P2, the solution z_g can be used for generating the optimal group combination \mathcal{C}_{opt} by deleting the unscheduled groups ($z_g = 0$) from \mathcal{G} . The optimal method is summarized as Algorithm 1.

Algorithm 1 Optimal Algorithm

- 1: **Inputs:**
 - 2: User demands $Q_k, k \in \mathcal{K}$
 - 3: Channel vectors $\mathbf{h}_k, k \in \mathcal{K}$
 - 4: Rate requirements $\eta_k, k \in \mathcal{K}$
 - 5: The power budget P_{tot}
 - 6: **for** each $g \in \mathcal{G}$ **do**
 - 7: Solve P1'(g) and obtain $p_k^g, k \in \mathcal{K}_g$
 - 8: Compute t_g based on $\max_{k \in \mathcal{K}_g} \frac{Q_k}{R_k^g}$
 - 9: **end for**
 - 10: Solve P2 and obtain $z_g, g \in \mathcal{G}$
 - 11: Use the groups with $z_g = 1$ to form a combination \mathcal{C}_{opt}
 - 12: **Outputs:**
 - 13: \mathcal{C}_{opt} and power allocation $p_k^g, k \in \mathcal{K}_g, g \in \mathcal{C}_{opt}$
-

By our characterizations, the optimal solution of P0 can be obtained by solving a convex problem for each group along with solving an integer linear programming problem P2. The whole problem is inherently hard and with high complexity since exponential number of groups need to be optimized by P1'(g). Moreover, the subproblem P2 is equivalent to an exact cover problem which is known as NP-complete [12]. Algorithm 1 can be used as an offline optimization method to provide optimal solutions, and for performance benchmarking. However, the searching space and the consumed time exponentially increase with the number of users, then this algorithm may not be able to meet the stringent latency requirements in real-time applications.

IV. SCHEDULING ALGORITHM DESIGN BASED ON DEEP LEARNING

In order to overcome the high complexity in obtaining the optimum, we design a deep neural network (DNN) based

approach to provide a high-quality and computational-efficient solution for the primal problem P0. The optimal algorithm in section III is used to generate training sets. We then train a DNN which is expected to map from the problem instances to the corresponding optimal scheduling solutions. The well-trained DNN model is applied to predict the scheduled groups. In general, the DNN outputs can not be treated as the optimal solution directly, since P0 has an exponential number of variables and has constraints to be satisfied. This introduces difficulties to achieve solution feasibility and good prediction performance. For instance, DNN may determine to schedule several groups. These groups could violate constraints (4d).

By our design, DNN is expected to provide a learning model to identify the relationship between the system inputs and the optimal group combinations, which could further help us to derive a scheduling strategy towards the goal of minimizing transmission time. By analyzing the DNN results, the groups with low probability to be scheduled will be deleted from the candidate list \mathcal{G} . After that, the remaining groups need to be further processed by a re-selection mechanism to make the final decision.

In the training phase, DNN establishes a prediction model to learn the relationship between the inputs, e.g., channel realizations, demands, and the optimal scheduling decisions. We denote the training set as (\mathbf{x}, \mathbf{y}) . The input \mathbf{x} refers to the channel vector $\mathbf{h}_k \in \mathbb{C}^{L \times 1}$, users' demand Q_1, \dots, Q_K , users' rate requirement η_1, \dots, η_K , and the maximum power P_{tot} . In order to avoid a huge number of output nodes, the optimization variables z_1, \dots, z_G are not used as the output \mathbf{y} directly. Instead, the output node is designed as the possibility of using a group combination. The reason is that due to the system limitations, e.g., number of antennas, the number of all the group combinations, denoted by N , can be much less than the number of groups G . Moreover, the feasibility issue of constraints (4d) can be resolved in each combination. By our design, we first list all of the candidate combinations as a union $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_N\}$. Then the output of the DNN prediction is organized in a N -dimension binary vector $\mathbf{y} = [y_1, \dots, y_N]$. If $y_n = 1$, it means the combination \mathcal{C}_n is selected and all the groups in combination \mathcal{C}_n are scheduled. For example, when $K = 3$ and $L = 2$, the candidate groups are $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}$ and $G = 6$. Note that the group $\{1, 2, 3\}$ is excluded due to the practical limit of $|\mathcal{K}_g| \leq L$. The candidate combinations are $\{\{1\}, \{2\}, \{3\}\}, \{\{1, 2\}, \{3\}\}, \{\{1, 3\}, \{2\}\}, \{\{2, 3\}, \{1\}\}$ and $N = 4$. If the combination $\{\{1, 2\}, \{3\}\}$ is selected, vector \mathbf{y} is $[0, 1, 0, 0]$ in the training set.

After sufficient training, the DNN is able to predict the grouping information \mathbf{y} . That is, given a new input $\hat{\mathbf{x}}$, the DNN will provide the estimated scheduling decision $\hat{\mathbf{y}}$. However, the original DNN output is not binary which cannot serve as an indicator for user scheduling. To solve the problem, we use the sigmoid function [13] in the DNN's output layer to limit the value between 0 and 1. Then, a rounding operation is adopted to convert fractional values to binary. More specifically, we use M as the mean of $\hat{\mathbf{y}}$. If any fractional value $\hat{y}_n > \alpha M$,

we set $\hat{y}_n = 1$, otherwise zero, where $\alpha > 0$ is a control parameter.

It is worth nothing that there are three cases with regard to $\hat{\mathbf{y}}$ after rounding. The ideal case is $\|\hat{\mathbf{y}}\| = 1$, which means only one element is 1. Then the scheduled groups are from this only combination. In the second case, vector $\hat{\mathbf{y}}$ may have multiple elements with value "1". Hence, we design a re-selection scheme to make final decisions. Firstly, we delete a considerably large amount of combinations from \mathcal{C} according to $\hat{\mathbf{y}}$. The remaining combinations compose a reduced candidate list \mathcal{C}^* . For example, if $\hat{\mathbf{y}} = [0, 1, 1, 0]$, then the combination list $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$ is reduced to $\mathcal{C}^* = \{\mathcal{C}_2, \mathcal{C}_3\}$. After that, we recall Algorithm 1 with the restricted set \mathcal{C}^* and obtain the final scheduling combination \mathcal{C}_{dnn} more efficiently. In the third case, DNN may return an all-zero vector $\hat{\mathbf{y}}$. For dealing with this case, we then adjust the control parameter α as follows to enable at least one element with value "1".

Algorithm 2 Learning-based Approach

- 1: **Inputs:**
 - 2: All the available combinations \mathcal{C}
 - 3: Requested data Q_k from each user
 - 4: Channel vector \mathbf{h}_k for each user
 - 5: Rate requirements η_1, \dots, η_K
 - 6: Power budget P_{tot}
 - 7: Given a new input $\hat{\mathbf{x}}$ to the well-trained DNN
 - 8: Apply rounding operations with α then obtain $\hat{\mathbf{y}}$
 - 9: **if** $\|\hat{\mathbf{y}}\| > 1$ **then**
 - 10: Delete combinations from \mathcal{C} according to the zero elements in $\hat{\mathbf{y}}$
 - 11: Obtain the restricted set \mathcal{C}^*
 - 12: Apply Algorithm 1 based on \mathcal{C}^*
 - 13: Obtain the selected combination \mathcal{C}_{dnn} and power
 - 14: allocation for each user
 - 15: **else**
 - 16: **if** $\|\hat{\mathbf{y}}\| = 1$ **then**
 - 17: Choose the only combination as \mathcal{C}_{dnn}
 - 18: **else if** $\|\hat{\mathbf{y}}\| = 0$ **then**
 - 19: Reduce α until $\|\hat{\mathbf{y}}\| \geq 1$
 - 20: **end if**
 - 21: **for** each group g in \mathcal{C}_{dnn} **do**
 - 22: solve the power control problem P1'(g) and obtain p_k^g
 - 23: **end for**
 - 24: **end if**
 - 25: **Outputs:**
 - 26: \mathcal{C}_{dnn} and power allocation for each user
-

If the well-trained DNN is able to predict the decisions precisely, the computation time can be reduced dramatically without loss any optimality. In case the prediction $\hat{\mathbf{y}}$ is not completely precise, the control parameter α is able to improve the accuracy. By our design, smaller α increases the likelihood of "1" in $\hat{\mathbf{y}}$, which also contributes to the improvement of prediction accuracy. On the other hand, the more elements

with value “1” involved in $\hat{\mathbf{y}}$, the more searching time required, since we need to solve $P1'(g)$ for more groups in \mathcal{C}^* . Therefore, there is a trade-off between prediction accuracy and computational complexity by adjusting α . The DNN-based approach is summarized in Algorithm 2.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the DNN-assisted approach, in terms of computational time, and performance gaps between the proposed learning approach and the optimal solutions. In the simulation, the BS is equipped with up to 5 antennas, serving up to 10 users. We limit the cardinality of each group to the number of antennas. If $L = 5$ and $K = 10$, the number of candidate groups is $G = C_{10}^5$ and the number of the group combinations is $N = \frac{C_{10}^5}{2}$. We design a fully connected DNN with two hidden layers. The DNN input is organized to a one-dimension vector, which consists of channel coefficients $|h_{lk}|^2, \forall l \in \{1, \dots, L\}, \forall k \in \{1, \dots, K\}$, users’ demand Q_1, \dots, Q_K , users’ rate requirement η_1, \dots, η_K , and power limit P_{tot} . Thus the number of the input nodes is $K(L + 2) + 1$. After training, in the DNN testing phase, we use 100 test sets/samples to average the results. The parameter settings for the DNN are summarized in Table I.

Table I
DNN SETTINGS

Parameter	Value
Number of input nodes	$K(L + 2) + 1$
Hidden layer-1 nodes	100
Hidden layer-2 nodes	80
Number of output nodes	N
Active function in hidden layers	Relu
Active function in output layer	Sigmoid
Optimizer	Adam optimization [13]

A. Comparison in Computation Time

To demonstrate the computation efficiency of the DNN-assisted approach, the CPU time (in seconds) of the DNN and the optimal method are compared in Table II. According to Algorithm 1, we optimally solve the power control problems $P1'(g)$ for all the groups in \mathcal{G} by applying the CVX solver. The complexity exponentially increases with the number of users K . For the DNN-assisted approach, we evaluate two types

Table II
CPU TIME IN COMPUTATION

Algorithmic Solutions	$K = 4$ $L = 2$	$K = 10$ $L = 5$
Algorithm 1	6.031	354.140
DNN test phase in Alg. 2	0.083	0.139
Alg. 2 for case 1, $\ \hat{\mathbf{y}}\ = 1$	1.211	2.910
Alg. 2 for case 2, $\ \hat{\mathbf{y}}\ > 1$	2.451	32.145
Alg. 2 for case 3, $\ \hat{\mathbf{y}}\ = 0$	1.330	3.039

of computational time consumption, i.e., the time consumed in DNN test phase (CPU time for executing lines 7-8 in Algorithm 2) and the total time consumption of Algorithm 2 (CPU time for executing line 7-26 in Algorithm 2). The former elapses from the moment that the well-trained DNN receives a new test set $\hat{\mathbf{x}}$ until returns the predicted vector $\hat{\mathbf{y}}$. The latter lasts from the moment of obtaining the prediction $\hat{\mathbf{y}}$ till the final decision \mathcal{C}_{dnn} is obtained. From Table I, the computational time in DNN test phase is insensitive to the problem’s scale. It keeps at the same magnitude in both cases. In general, the computational time of the proposed DNN-assisted approach (Algorithm 2) is dramatically reduced compared with the optimal algorithm. We remark that the total CPU time in Algorithm 2 varies with three cases of $\|\hat{\mathbf{y}}\|$. It is noted that, in the case of $\|\hat{\mathbf{y}}\| > 1$, the time is more than the other two cases. This is because when $\|\hat{\mathbf{y}}\| > 1$, the power allocation problem $P1'(g)$ needs to be solved for more groups.

B. Comparison in Prediction Accuracy and Approximating Optimum

Fig. 1 shows the performance of the DNN-assisted approach compared with the optimal solution, with increasing the training set size. The performance gaps on the vertical axis represents the relative delivery time of the DNN-assisted method with respect to the optimum value. In general, using more data in the training set improves the prediction accuracy of the DNN-based method. In particular, the performance reaches to the optimum when the size of training set is around 800, leading to the smallest gap around 8%, 9% and 10% in the case of $\alpha = 2.4$, $\alpha = 2.6$ and $\alpha = 2.8$ respectively.

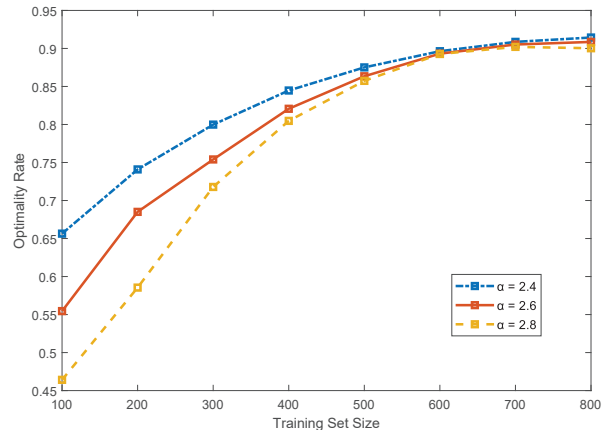


Figure 1. Average gaps between DNN and optimum ($K = 10, L = 5$)

Next, we use Fig. 2 to depict the difference of the total data transmission length T_{tot} between the DNN-assisted approach and the optimal method in each tested cases. Over 100 testing samples, the average gaps of objective values between the two algorithms are small, around 8%. This shows that the DNN approach is able to provide a near-optimal solution with high computational efficiency.

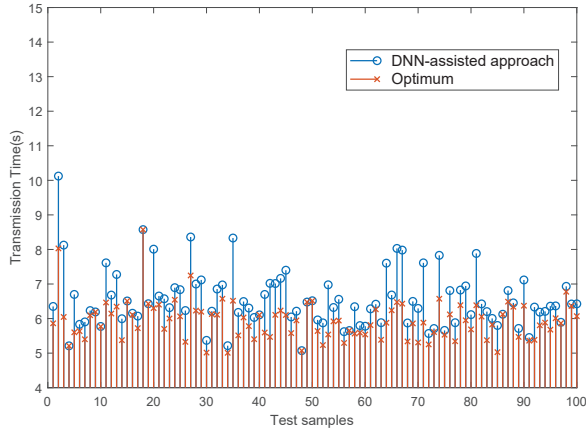


Figure 2. Transmission time comparison between the DNN-based algorithm and the optimal algorithm ($K = 10$, $L = 5$, $\alpha = 2.8$, training set size= 800)

C. Trade-off Between Optimality and Computational Time

Next, we evaluate the effect of α on the performance gap and the computation complexity in the re-selection phase. As shown in Fig. 3, the time monotonically decreases with the growth of α , whereas the performance of the DNN in approaching to the optimum is degraded. As we analyzed in section IV, when α is large, a small amount of combinations will be in the restricted candidate set \mathcal{C}^* , thus leading to less computational time. On the other hand, the fewer candidates in \mathcal{C}^* can possibly result in higher the probability to loss optimality.

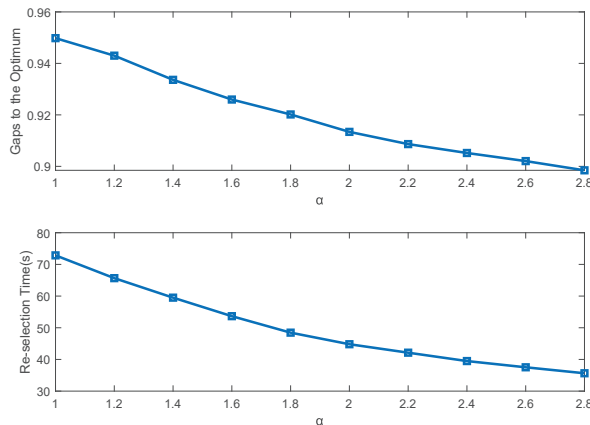


Figure 3. Optimality and re-selection time with different α ($K = 10$, $L = 5$, training set size= 800)

VI. CONCLUSION

For minimizing the data delivery time, we formulated a joint user scheduling and power allocation problem for MISO systems. In order to obtain the global optimum, an optimal method, with exponential complexity, was designed. Toward an efficient solution, we trained a DNN to learn the mapping

function between the problem's inputs and the optimal solutions. We proposed a DNN-assisted approach to enable near-optimality and less computational delay. Numerical results show the DNN-assisted approach can well-approximate the optimum with low computation complexity.

VII. ACKNOWLEDGMENT

The work has been supported by the European Research Council (ERC) project AGNOSTIC, the FNR bilateral projects LARGOS, and the FNR CORE projects ROSETTA (11632107) and ProCAST.

REFERENCES

- [1] M. Agiwal, A. Roy and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, third quarter 2016.
- [2] V. Angelakis, A. Ephremides, Q. He and D. Yuan, "Minimum-time link scheduling for emptying wireless systems: solution characterization and algorithmic framework," in *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1083–1100, Feb. 2014.
- [3] L. Lei, D. Yuan, C. K. Ho, and S. Sun, "Optimal cell clustering and activation for energy saving in load-coupled wireless networks," in *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6150–6163, Nov. 2015.
- [4] S. Lahoud, K. Khawam, S. Martin, G. Feng, Z. Liang and J. Nasreddine, "Energy-Efficient Joint Scheduling and Power Control in Multi-Cell Wireless Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3409–3426, Dec. 2016.
- [5] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier and A. Ephremides, "Minimum-energy link scheduling for emptying wireless networks," in *proc. 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 207–212, 2015.
- [6] K. Rahimi Malekshan and W. Zhuang, "Joint Scheduling and Transmission Power Control in Wireless Ad Hoc Networks," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5982–5993, Sept. 2017.
- [7] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
- [8] Z. Chang, L. Lei, Z. Zhou, S. Mao and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," in *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, June 2018.
- [9] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled HetNet," in *proc. IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2017.
- [10] 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 International Conference on Communications (ICC)*, 2018.
- [11] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-Caching Wireless Networks: Performance analysis and optimization," in *IEEE Transaction on Wireless Communication*, vol. 17, no. 7, pp. 2827 – 2839, Apr. 2018.
- [12] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.