Neural Combinatorial Optimization for Throughput Maximization in IRS-Aided Systems

Rui Huang and Vincent W.S. Wong
Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
email: {ruihuang, vincentw}@ece.ubc.ca

***Abstract*—Intelligent reflecting surface (IRS) is a promising paradigm for enhancing the spectrum efficiency of wireless communication systems. In this paper, we study the joint uplink scheduling and phase shift control in IRS-aided systems. We formulate the throughput maximization problem as a combinatorial optimization problem. We decompose the problem into two subproblems for user scheduling and phase shift control, respectively. We propose a neural combinatorial optimization (NCO)-based algorithm, in which a near-optimal stochastic policy for user scheduling is learned by deep neural networks (DNNs) with attention mechanism, while the phase shifts of the IRS are optimized using fractional programming. Unlike alternating optimization-based approaches which obtain a suboptimal solution by iteratively solving two subproblems, the proposed NCO-based algorithm is capable of obtaining a near-optimal solution while each subproblem is required to be solved only once. Simulation results show that the proposed NCO-based algorithm achieves an aggregate throughput which is within 98% of the exhaustive search algorithm, and outperforms both greedy scheduling and random scheduling algorithms.**

I. Introduction

Intelligent reflecting surface (IRS) is a reconfigurable planar surface with multiple passive reflecting elements. Each element on the IRS can perform a phase shift to the incident signal independently and reflect the shifted signal to a receiver. An IRS-aided system serving three users to perform uplink transmissions is shown in Fig. 1. Apart from the direct channels between the base station and the user equipment, IRS introduces additional propagation channels. When the line-of-sight (LOS) links between the base station and users are blocked by obstacles, deploying an IRS can create virtual LOS channels to facilitate data transmission and improve the coverage of the base station. Moreover, by properly control the phase shift of the reflecting elements on IRS, the base station can mitigate interference and allow multiple users to share one physical resource block (PRB) to perform uplink transmissions.

Existing research on resource allocation in IRS-aided systems mostly focus on the optimization of the beamforming at the base station and the phase shift control of IRS [1]–[5]. The beamforming and phase shift optimization for IRS-aided systems with single user have been studied in [1], [2]. Recently, beamforming and phase shift design for interference mitigation in IRS-aided systems with multiple users have been proposed in [4], [5]. The authors in [4] studied the ergodic rate of an IRS-aided system with interference from a secondary user and proposed a parallel coordinate descent-based algorithm to optimize the phase shift. The authors in [5] investigated an IRS-aided system with multiple primary and secondary users, and proposed an alternating optimization (AO)-based algorithm to jointly optimize the beamforming vector and phase-shift matrix. However, while the aforementioned algorithms are designed for a given set of users, the uplink scheduling problem in IRS-aided systems has not been investigated. For an IRS-aided system with multiple users, it is necessary and beneficial for the base station to properly schedule the transmissions of the users, such that the potential interference between the users can be mitigated. Moreover, in the existing AO-based approaches, the iterative optimization process has to be invoked again whenever the base station observes a change in the channel states. This may lead to a high computational complexity.

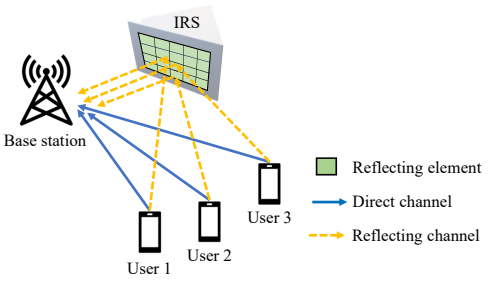

Fig. 1. An IRS-aided system with three users sharing one uplink PRB. The direct channels are denoted by blue solid lines, while the reflecting channels are denoted by yellow dashed lines.

To tackle these challenges, in this paper, we propose a neural combinatorial optimization (NCO)-based algorithm for maximizing the aggregate throughput by jointly optimizing the uplink scheduling and phase shift control of an IRS-aided system. NCO is a powerful tool for solving combinatorial optimization problems, where the optimal solution is a permutation of the optimization variables [6]–[8]. In our proposed NCO-based approach, we apply reinforcement learning and train the deep neural networks (DNNs) with *attention mechanism* [9] to obtain a stochastic policy. The proposed NCO-based algorithm is capable of obtaining a near-optimal solution while each subproblem is required to be solved only once. Our contributions are as follows:

- We formulate the joint user scheduling and phase shift optimization problem for aggregate throughput maximization in IRS-aided systems as a mixed-integer nonlinear optimization problem.

- We propose an NCO-based algorithm to learn the stochastic policy for obtaining a near-optimal solution. We employ two DNN modules, i.e., an encoder module and a decoder module. The encoder module learns the high-dimensional representations of the channel information of the users, and these representations are used by the decoder module to obtain the stochastic policy.
- We propose an offline training algorithm, in which the DNNs are trained based on reinforcement learning without requiring the optimal solution of the problem during the training phase. The learned stochastic policy can be applied to solve the problem in an online manner.
- Simulation results show that the proposed NCO-based algorithm achieves an aggregate throughput that is within 98% of the exhaustive search algorithm and outperforms greedy scheduling and random scheduling algorithms.

This paper is organized as follows. The system model and problem formulation are presented in Section II. The NCO-based algorithm is given in Section III. Simulation results are shown in Section IV. Conclusions are drawn in Section V.

II. System Model

We consider an IRS-aided system with one base station, an IRS, and N users. The set of users is denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. The base station and the users are equipped with one antenna. Time is slotted into intervals of equal duration. The time interval $[t, t+1)$ is referred to as time slot t, where $t \in \mathcal{T} = \{0, 1, 2, \ldots, T-1\}$. An IRS with L_R phase-shifting elements is deployed to facilitate the uplink transmission of the users. In each time slot $t \in \mathcal{T}$, the base station schedules M users to perform uplink transmission using one PRB. Similar settings have also been adopted in [3]. We use binary control variable $x_n(t) \in \{0, 1\}$ to indicate whether user $n \in \mathcal{N}$ is scheduled for uplink transmission in time slot t. We set $x_n(t) = 1$ if user n is scheduled in time slot t, and $x_n(t) = 0$ otherwise. In time slot t, we have

$$x_n(t) \in \{0, 1\},\ n \in \mathcal{N}, \tag{1}$$

$$\sum_{n \in \mathcal{N}} x_n(t) = M. \tag{2}$$

We use vector $\boldsymbol{x}(t) = (x_1(t),\, x_2(t),\, \ldots,\, x_n(t))$ to denote all control variables $x_n(t)$ in time slot t.

Let $h_{D,n}(t) \in \mathbb{C}$ and $\boldsymbol{h}_{R,n}(t) \in \mathbb{C}^{L_R}$ denote the channel gain between user $n \in \mathcal{N}$ and the base station, and the channel gain between user n and the IRS in time slot $t \in \mathcal{T}$, respectively. The channel gain between the IRS and the base station in time slot t is denoted by $\boldsymbol{g}(t) \in \mathbb{C}^{L_R}$. We assume perfect channel estimation at the base station. We use diagonal matrix $\boldsymbol{\Psi}(t)$ to denote an $L_R \times L_R$ diagonal phase-shift matrix of the IRS in time slot t. We have

$$\boldsymbol{\Psi}(t) = \text{diag}(e^{j\psi_1(t)}, \cdots, e^{j\psi_{L_R}(t)}) \in \mathbb{C}^{L_R \times L_R}, \tag{3}$$

where $\psi_l(t),\, l \in \{1, \ldots, L_R\}$ is the phase shift of the l-th reflecting element on the IRS. We have the following constraint on the phase shift of each reflecting element

$$\psi_l(t) \in [0,\, 2\pi),\, l \in \{1, \ldots, L_R\}. \tag{4}$$

We assume the scheduled users always use the maximum transmit power $P^{\max}$ for packet transmission. The received signal of user n at base station in time slot $t \in \mathcal{T}$ is given by

$$y_n(t) = x_n(t)\sqrt{P^{\max}}\,\big(h_{D,n}(t)\, s_n(t) + \boldsymbol{g}^H(t)\, \boldsymbol{\Psi}(t)\, \boldsymbol{h}_{R,n}(t)\, s_n(t)\big) + I_n(t) + w,$$

where $(\cdot)^H$ denotes the conjugate transpose, $s_n(t) \in \mathbb{C}$ is the symbol of user n in time slot t with unit power, w is the complex Gaussian noise with zero mean and variance σ^2, and $I_n(t)$ is the interference from the remaining scheduled users in time slot t. $I_n(t)$ is given by

$$I_n(t) = \sum_{j \in \mathcal{N} \setminus \{n\}} x_j(t)\, \sqrt{P^{\max}}\big(h_{D,j}(t)\, s_j(t) + \boldsymbol{g}^H(t)\, \boldsymbol{\Psi}(t)\, \boldsymbol{h}_{R,j}(t)\, s_j(t)\big).$$

The signal-to-interference-plus-noise ratio (SINR) of user n in time slot t is given by

$$\Gamma_n(t) = \frac{x_n(t) P^{\max} \left| h_{D,n}(t) + \boldsymbol{g}^H(t) \boldsymbol{\Psi}(t) \boldsymbol{h}_{R,n}(t) \right|^2}{\sum\limits_{j \in \mathcal{N} \setminus \{n\}} x_j(t) P^{\max} \left| h_{D,j}(t) + \boldsymbol{g}^H(t) \boldsymbol{\Psi}(t) \boldsymbol{h}_{R,j}(t) \right|^2 + \sigma^2}.$$

The achievable throughput (bits/time slot/Hz) of user n in time slot t can be determined as follows

$$R_n(\boldsymbol{x}(t), \boldsymbol{\Psi}(t)) = \log_2(1 + \Gamma_n(t)).$$

We formulate the following aggregate throughput maximization problem by jointly optimizing the uplink scheduling of the users and the phase-shift matrix of the IRS in each time slot $t \in \mathcal{T}$:

$$\begin{aligned} \underset{\boldsymbol{x}(t),\, \boldsymbol{\Psi}(t)}{\text{maximize}} \quad & \sum_{n \in \mathcal{N}} R_n(\boldsymbol{x}(t), \boldsymbol{\Psi}(t)) \\ \text{subject to} \quad & \text{constraints } (1), (2), (4). \end{aligned} \tag{5}$$

Problem (5) is a mixed-integer nonlinear optimization problem due to the binary control variables $\boldsymbol{x}(t)$ and the fractional objective function. In time slot $t \in \mathcal{T}$, problem (5) can be decomposed into two subproblems for the user scheduling and phase-shift control, respectively. In particular, given the phase-shift matrix $\boldsymbol{\Psi}(t)$, the subproblem for user scheduling in time slot t is as follows:

$$\begin{aligned} \underset{\boldsymbol{x}(t)}{\text{maximize}} \quad & \sum_{n \in \mathcal{N}} R_n(\boldsymbol{x}(t)) \\ \text{subject to} \quad & \text{constraints } (1) \text{ and } (2). \end{aligned} \tag{6}$$

Problem (6) is a combinatorial optimization problem, in which we have in total $\binom{N}{M}$ feasible user scheduling selections in each time slot. This problem is NP-complete and the optimal solution is difficult to obtain. Given $\boldsymbol{x}(t)$, the second subproblem for phase-shift matrix optimization is as follows:

$$\begin{aligned} \underset{\boldsymbol{\Psi}(t)}{\text{maximize}} \quad & \sum_{n \in \mathcal{N}} R_n(\boldsymbol{\Psi}(t)) \\ \text{subject to} \quad & \text{constraint } (4). \end{aligned} \tag{7}$$

Subproblem (7) can be transformed into a multi-ratio fractional programming problem [10] by using semidefinite relaxation

to tackle constraint (4). The details of solving subproblem (7) are shown in the Appendix. Based on the decomposition, a suboptimal solution of problem (5) can be obtained by solving subproblem (7) for all feasible solutions of subproblem (6). The iterative process is required to be repeated whenever the base station obtains new channel realizations of the users. This iterative process is computational expensive. In the next section, we propose an NCO-based approach that can obtain a near-optimal solution for problem (5) with a lower computational complexity.

III. NCO-based Throughput Maximization for IRS-Aided Systems

In this section, we first introduce a general stochastic policy for solving problem (5). We then propose an NCO-based algorithm to efficiently learn a near-optimal stochastic policy.

A. Stochastic Policy for Uplink Scheduling

We use vector $\boldsymbol{v}_n(t)$ to collect the channel information of user n along with the channel gain between the IRS and the base station in time slot t. In particular, this *user-specific* vector $\boldsymbol{v}_n(t)$ for user n is given by

$$\boldsymbol{v}_n(t) = (h_{D,n}(t), \boldsymbol{h}_{R,n}(t), \boldsymbol{g}(t)),\ n \in \mathcal{N}. \tag{8}$$

We use set $\mathcal{V}(t) = \{\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_N(t)\}$ to collect the user-specific vectors in time slot t. Problem (5) in time slot t can be solved with the following steps:

Step 1: Find a subset $\mathcal{U}(t)$ which consists of M different vectors from set $\mathcal{V}(t)$. That is, find a subset $\mathcal{U}(t) = \{\boldsymbol{u}_1(t), \ldots, \boldsymbol{u}_M(t)\}$ such that $\boldsymbol{u}_l(t) \in \mathcal{V}(t)$ and $\boldsymbol{u}_l(t) \neq \boldsymbol{u}_{l'}(t)$, $\forall l \neq l', l, l' \in \{1, \ldots, M\}$. Given subset $\mathcal{U}(t)$, we obtain the corresponding user scheduling $\boldsymbol{x}(t)$ in time slot t by setting $x_n(t) = 1$ if $\boldsymbol{v}_n(t) \in \mathcal{U}(t)$. Otherwise, we set $x_n(t) = 0$. This is equivalent to finding a feasible solution of subproblem (6).

Step 2: After determining subset $\mathcal{U}(t)$, a reward $r(\mathcal{U}(t))$ is revealed. This reward is given by the maximum aggregate throughput, i.e., $r(\mathcal{U}(t)) = \max_{\boldsymbol{\Psi}(t)} \sum_{n\in\mathcal{N}} R_n(\boldsymbol{\Psi}(t))$ subject to constraint (4), with the user scheduling specified by the subset $\mathcal{U}(t)$. This is equivalent to solving subproblem (7).

Note that the optimal solution of problem (5) can be obtained by finding the subset $\mathcal{U}(t)$ of $\mathcal{V}(t)$ with the maximum reward $r(\mathcal{U}(t))$. In time slot $t \in \mathcal{T}$, the stochastic policy for solving such problem can be defined by the conditional probability of selecting a particular subset $\mathcal{U}(t)$ under given $\mathcal{V}(t)$, i.e., $p(\mathcal{U}(t) \,|\, \mathcal{V}(t))$. Using the chain rule, this probability can be factorized as follows [7]

$$p(\mathcal{U}(t) \,|\, \mathcal{V}(t)) = \prod_{l=1}^{M} p(\boldsymbol{u}_l(t) \,|\, \mathcal{V}(t), \boldsymbol{u}_1(t), \ldots, \boldsymbol{u}_{l-1}(t)). \tag{9}$$

The stochastic policy in (9) shows that, to determine the user scheduling in time slot t, the base station selects M different vectors from set $\mathcal{V}(t)$ sequentially. That is, the base station selects the first vector based on set $\mathcal{V}(t)$ and then selects the second vector based on set $\mathcal{V}(t)$ and the previously selected vector. This process is repeated until $\mathcal{U}(t)$ contains M vectors.

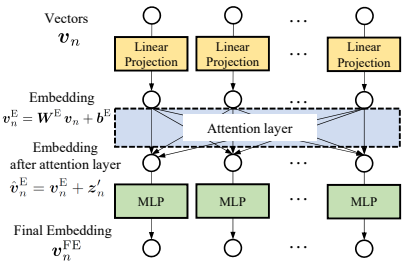

Fig. 2. The network structure of the encoder module. The encoder module learns the embeddings of the vectors by passing the channel information through a linear projection layer, an attention layer [9], and an MLP layer.

B. NCO-based Algorithm: Encoder Module

Using NCO, the stochastic policy in (9) for obtaining the maximum expected reward can be learned and parameterized by the DNN modules with learnable parameters $\boldsymbol{\Phi}$. The parameterized policy is denoted by $p_{\boldsymbol{\Phi}}(\mathcal{U} \,|\, \mathcal{V})$. As we aim to learn the generalized policy that can be applied to solve problem (5) with any possible channel realizations in any time slots, we drop the notation for time slot t.

To obtain $p_{\boldsymbol{\Phi}}(\boldsymbol{u}_l \,|\, \mathcal{V}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{l-1})$, the DNN module takes the channel information of the users as input and approximate the desired conditional probability. To this end, we first use an *encoder* DNN module to learn the underlying structures and abstractions of the channel information, which are referred to as the *embedding* of the users [7]. The DNN structure for the encoder module is shown in Fig 2. For vector $\boldsymbol{v}_n \in \mathcal{V}$, its d_h-dimensional embedding $\boldsymbol{v}_n^{\mathrm{E}}$ can be determined based on the following linear projection:

$$\boldsymbol{v}_n^{\mathrm{E}} = \boldsymbol{W}^{\mathrm{E}}\,\boldsymbol{v}_n + \boldsymbol{b}^{\mathrm{E}}, \tag{10}$$

where the weights $\boldsymbol{W}^{\mathrm{E}} \in \mathbb{R}^{d_h \times d_i}$ and biases $\boldsymbol{b}^{\mathrm{E}} \in \mathbb{R}^{d_h \times d_i}$ are learnable parameters, d_i is the size of $\boldsymbol{v}_n$, and d_h is a constant. After linear projection in (10), we use the *attention mechanism* [9] to capture the inter-user interference and the combinatorial structure of the optimization problem. The attention mechanism can be considered as an information exchange process between the embeddings of vectors, such that the embedding of a particular vector not only represents its own channel features, but also reflects how it relates to the other vectors. This step is important for the DNN module to learn about the interference between the users. To this end, we generate three additional vectors, namely, *key* $\boldsymbol{k}_n$, *query* $\boldsymbol{q}_n$, *value* $\boldsymbol{z}_n$, for each vector based on its embedding as follows:

$$\boldsymbol{k}_n = \boldsymbol{W}_{\mathrm{en}}^{\mathrm{K}}\,\boldsymbol{v}_n^{\mathrm{E}},\ \boldsymbol{q}_n = \boldsymbol{W}_{\mathrm{en}}^{\mathrm{Q}}\,\boldsymbol{v}_n^{\mathrm{E}},\ \boldsymbol{z}_n = \boldsymbol{W}_{\mathrm{en}}^{\mathrm{Z}}\,\boldsymbol{v}_n^{\mathrm{E}}, \tag{11}$$

where matrices $\boldsymbol{W}_{\mathrm{en}}^{\mathrm{K}}, \boldsymbol{W}_{\mathrm{en}}^{\mathrm{Q}} \in \mathbb{R}^{d_k \times d_h}$ and $\boldsymbol{W}_{\mathrm{en}}^{\mathrm{Z}} \in \mathbb{R}^{d_z \times d_h}$ are learnable parameters, and d_k, d_z are constants. Using the attention mechanism, the embedding of a vector may receive values from the embeddings of other vectors. To determine the value that embedding $\boldsymbol{v}_n^{\mathrm{E}}$ received from embedding $\boldsymbol{v}_j^{\mathrm{E}}$, we compute the *compatibility* $\delta_{n,j} \in \mathbb{R}$ of the two vectors based on the query $\boldsymbol{q}_n$ of $\boldsymbol{v}_n^{\mathrm{E}}$ and the key $\boldsymbol{k}_j$ of $\boldsymbol{v}_j^{\mathrm{E}}$:

$$\delta_{n,j} = \frac{\boldsymbol{q}_n^{\mathrm{T}}\boldsymbol{k}_j}{\sqrt{d_k}}. \tag{12}$$

The attention weights $a_{n,j} \in [0,\, 1]$ can be obtained using the following softmax function:

$$a_{n,j} = \frac{e^{\delta_{n,j}}}{\sum_{j' \in \mathcal{N}} e^{\delta_{n,j'}}}. \tag{13}$$

The values that $\boldsymbol{v}_n^{\mathrm{E}}$ received from the embeddings of other vectors can be determined as follows:

$$\boldsymbol{z}'_n = \sum_{j \in \mathcal{N}} a_{n,j} \boldsymbol{z}_j. \tag{14}$$

We construct a new embedding $\hat{\boldsymbol{v}}_n^{\mathrm{E}}$ of vector $\boldsymbol{v}_n$ by combining the original embedding $\boldsymbol{v}_n^{\mathrm{E}}$ with the received values $\boldsymbol{z}'_n$:

$$\hat{\boldsymbol{v}}_n^{\mathrm{E}} = \boldsymbol{v}_n^{\mathrm{E}} + \boldsymbol{z}'_n. \tag{15}$$

The final embedding of vector $\boldsymbol{v}_n$, which is denoted by $\boldsymbol{v}_n^{\mathrm{FE}}$, is obtained by passing the embedding $\hat{\boldsymbol{v}}_n^{\mathrm{E}}$ through a multilayer perceptron (MLP) module. The MLP module has one hidden layer with dimension d_e. We denote the learnable parameters of the MLP module as $\boldsymbol{\Phi}^{\mathrm{MLP}}$. The aggregate embedding of all vectors in $\mathcal{V}$ is determined by the average of the final embeddings of all vectors, which is [7]

$$\boldsymbol{v}_G^{\mathrm{E}} = \frac{1}{N} \sum_{n \in \mathcal{N}} \boldsymbol{v}_n^{\mathrm{FE}}. \tag{16}$$

The parameters in the encoder module $\boldsymbol{\Phi}_{\mathrm{en}}$ is given by

$$\boldsymbol{\Phi}_{\mathrm{en}} = (\boldsymbol{W}^{\mathrm{E}}, \boldsymbol{b}^{\mathrm{E}}, \boldsymbol{W}_{\mathrm{en}}^{\mathrm{K}}, \boldsymbol{W}_{\mathrm{en}}^{\mathrm{Q}}, \boldsymbol{W}_{\mathrm{en}}^{\mathrm{Z}}, \boldsymbol{\Phi}^{\mathrm{MLP}}). \tag{17}$$

The outputs of the encoder module are the final embeddings of the vectors as given by (15) and (16).

C. NCO-based Algorithm: Decoder Module

We employ another DNN module called the *decoder* module to generate the stochastic policy based on the final embeddings provided by the encoder module. The decoder module maintains a *context embedding*. The motivation of introducing the context embedding is to account for the conditions in (9) when generating the conditional probabilities. In particular, the context embedding for generating the conditional probability $p_{\boldsymbol{\Phi}}(\boldsymbol{u}_l \,|\, \mathcal{V}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{l-1})$ is given by

$$\boldsymbol{v}_c^{\mathrm{E}} = \begin{cases} [\boldsymbol{v}_G^{\mathrm{E}},\, \boldsymbol{v}_{\boldsymbol{u}_0},\, M\,], & \text{if } \; l = 1, \\ [\boldsymbol{v}_G^{\mathrm{E}},\, \boldsymbol{v}_{\boldsymbol{u}_{l-1}}^{\mathrm{FE}},\, M - l + 1\,], & \text{if } \; l > 1, \end{cases} \tag{18}$$

where $[\cdot,\, \cdot,\, \cdot]$ is the concatenation operator, $\boldsymbol{v}_{\boldsymbol{u}_{l-1}}^{\mathrm{FE}}$ is the embedding of the previously selected vector. $\boldsymbol{v}_{\boldsymbol{u}_0}$ serves as a placeholder to maintain the constant size of $\boldsymbol{v}_c^{\mathrm{E}}$ when the base station aims to determine the first scheduled user. The last element in (18), i.e., $M - l + 1$, is the number of remaining vectors that can be added into the subset.

To obtain the stochastic policy in the decoder module, we compute the compatibility of the context embedding in (18) with each of the remaining vectors that can be potentially added into the subset. We obtain the query of the context embedding, the keys and values of the vectors as follows:

$$\boldsymbol{q}_c = \boldsymbol{W}_{\mathrm{de}}^{\mathrm{Q}} \boldsymbol{v}_c^{\mathrm{E}}, \;\; \boldsymbol{k}_n = \boldsymbol{W}_{\mathrm{de}}^{\mathrm{K}} \boldsymbol{v}_n^{\mathrm{FE}}, \;\; \boldsymbol{z}_n = \boldsymbol{W}_{\mathrm{de}}^{\mathrm{Z}} \boldsymbol{v}_n^{\mathrm{FE}}, \tag{19}$$

where matrices $\boldsymbol{W}_{\mathrm{de}}^{\mathrm{Q}} \in \mathbb{R}^{d_h \times d_c}$, $\boldsymbol{W}_{\mathrm{de}}^{\mathrm{K}} \in \mathbb{R}^{d_h \times d_k}$, and $\boldsymbol{W}_{\mathrm{de}}^{\mathrm{Z}} \in \mathbb{R}^{d_h \times d_z}$ project the context embedding and the final embeddings of the vectors back to d_h dimensions. d_c is the size of $\boldsymbol{v}_c^{\mathrm{E}}$. These matrices are learned by the decoder module. The compatibilities of context embedding with the remaining vectors in $\mathcal{V}$ can now be determined by

$$\delta_{c,n} = \begin{cases} \alpha \tanh\left(\frac{\boldsymbol{q}_c^{\mathrm{T}} \boldsymbol{k}_n}{\sqrt{d_k}}\right), & \text{if } \boldsymbol{v}_n \text{ has not been selected}, \\ -\infty, & \text{otherwise}, \end{cases}$$

where α is a constant. We use $\alpha \tanh()$ to clip the compatibility within $[-\alpha,\, \alpha]$ to improve the performance [7]. The motivation of setting the compatibilities between the context embedding and the previously selected vectors to $-\infty$ is to prevent the base station from selecting duplicate vectors. The conditional probability for selecting vector $\boldsymbol{v}_n$ as the l-th vector in the subset is then given by

$$p(\boldsymbol{u}_l = \boldsymbol{v}_n \,|\, \mathcal{V}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{l-1}) = \frac{e^{\delta_{c,n}}}{\sum_{n' \in \mathcal{N}} e^{\delta_{c,n'}}}. \tag{20}$$

With the conditional probability in (20), the stochastic policy $p_{\boldsymbol{\Phi}}(\mathcal{U} \,|\, \mathcal{V})$ can be obtained based on (9). The learnable parameters in the decoder module is given by

$$\boldsymbol{\Phi}_{\mathrm{de}} = (\boldsymbol{W}_{\mathrm{de}}^{\mathrm{K}}, \boldsymbol{W}_{\mathrm{de}}^{\mathrm{Q}}, \boldsymbol{W}_{\mathrm{de}}^{\mathrm{Z}}). \tag{21}$$

The overall learnable parameters are collected as follows:

$$\boldsymbol{\Phi} = (\boldsymbol{\Phi}_{\mathrm{en}},\, \boldsymbol{\Phi}_{\mathrm{de}}). \tag{22}$$

D. Learning Algorithm Based on REINFORCE

The stochastic policy generated by the encoder and decoder modules is characterized by the learnable parameters $\boldsymbol{\Phi}$, i.e., $p_{\boldsymbol{\Phi}}(\mathcal{U} \,|\, \mathcal{V})$. To maximize the aggregate throughput, we need to find the parameters $\boldsymbol{\Phi}$, such that the expected reward of the subset, which is selected by policy $p_{\boldsymbol{\Phi}}(\mathcal{U} \,|\, \mathcal{V})$, is maximized. This leads to the following optimization problem:

$$\min_{\boldsymbol{\Phi}} \; \mathcal{L}(\boldsymbol{\Phi} \,|\, \mathcal{V}) \triangleq \mathbb{E}_{\mathcal{U} \sim p_{\boldsymbol{\Phi}}(\mathcal{U} \,|\, \mathcal{V})} \big[-r(\mathcal{U}) \big], \tag{23}$$

where $\mathcal{L}(\boldsymbol{\Phi} \,|\, \mathcal{V})$ is referred to as the *loss function*. To determine the loss function, we feed the vectors in $\mathcal{V}$ into the DNN modules and determine the subset $\mathcal{U}$. The reward $r(\mathcal{U})$ can be determined by solving subproblem (7) with the user scheduling given by $\mathcal{U}$. We then use the REINFORCE [11] algorithm to perform gradient descent and obtain the optimal learnable parameters $\boldsymbol{\Phi}$. The gradient is given by

$$\nabla \mathcal{L}(\boldsymbol{\Phi} \,|\, \mathcal{V}) = -\mathbb{E}_{\mathcal{U} \sim p_{\boldsymbol{\Phi}}(\mathcal{U} | \mathcal{V})} \big[r(\mathcal{U}) \nabla \log p_{\boldsymbol{\Phi}}(\mathcal{U} | \mathcal{V}) \big]. \tag{24}$$

With the gradient in (24), we use Adam optimizer [12] to solve problem (23) and update the learnable parameters $\boldsymbol{\Phi}$. This is referred to as the *training phase* of the DNN modules.

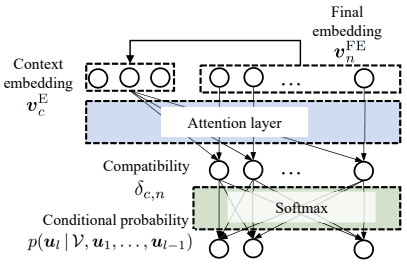

Fig. 3. The network structure of the decoder module. The decoder module generates the conditional probabilities based on the final embedding provided by the encoder module and the context embedding.

Algorithm 1 Training Algorithm in Each Training Iteration

1: Obtain a minibatch consisting of B different channel realizations of the users, and determine the corresponding vector set $\mathcal{V}_B$.
2: **for** each $\mathcal{V}' \in \mathcal{V}_B$ **do**
3: Feed the vectors in $\mathcal{V}'$ into the DNN modules and determine the subset $\mathcal{U}$.
4: Determine the reward $r(\mathcal{U})$ by solving subproblem (7) and obtain the loss function $\mathcal{L}(\mathbf{\Phi} \,|\, \mathcal{V}')$.
5: **end for**
6: Determine the aggregate loss over the minibatch as $\mathcal{L}(\mathbf{\Phi} \,|\, \mathcal{V}_B) = \frac{1}{B} \sum_{\mathcal{V}' \in \mathcal{V}_B} \mathcal{L}(\mathbf{\Phi} \,|\, \mathcal{V}')$.
7: Determine the gradient based on (24), and update $\mathbf{\Phi}$ by solving problem (23) using Adam optimizer [12].

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Dimensions of the attention layer d_h, d_k, d_z	256
Dimension of the hidden layer in the MLP module d_e	512
Learning rate in training phase	0.0001
Minibatch size B	512
Constant α for clipping the compatibility	10
Number of channel realizations used for evaluation	10000

The training algorithm is shown in Algorithm 1. During training, the DNN modules are applied to solve problem (5) with different channel realizations. In each training iteration, we train the DNN module using one minibatch, which consists of problem (5) under B different channel realizations. For each of the B channel realizations in the minibatch, we determine the corresponding vector set $\mathcal{V}'$. We use set $\mathcal{V}_B$ to collect the B vector sets in the minibatch. The loss function is first determined for each $\mathcal{V}' \in \mathcal{V}_B$, and then averaged over the whole minibatch. With the loss function, $\mathbf{\Phi}$ are updated based on (23) and (24). After training, the computation complexity of online execution of the proposed algorithm per time slot is $O(N^2 + C_{\text{FP}})$, where C_{FP} is the computational complexity of solving subproblem (7) using fractional programming.

IV. Performance Evaluation

We simulate an IRS-aided system where the distance between the IRS and the base station is 200 meters. The users are randomly and uniformly distributed within $[10, 150]$ meters of the IRS. We assume the channels between the users and the base station are blocked [3]. The reflecting channels follow Rician fading distribution. We set the Ricean-K factor to 6 [13]. The maximum transmit power $P^{\max}$ is set to 30 dBm, and the noise power is set to -90 dBm. The settings for NCO-based algorithm are shown in Table I. We compare with the following algorithms:

- Exhaustive search (ES)-based user scheduling with discretized phase-shift control: The base station iterates through all possible user scheduling selections. The phase shift control variables are discretized with step size $2\pi/20$ and ES is performed to obtain the maximum aggregate throughput.
- ES-based user scheduling with fractional programming (FP)-based phase-shift control: Apart from ES-based user scheduling, the base station obtains a suboptimal phase-shift matrix using FP as shown in the Appendix.

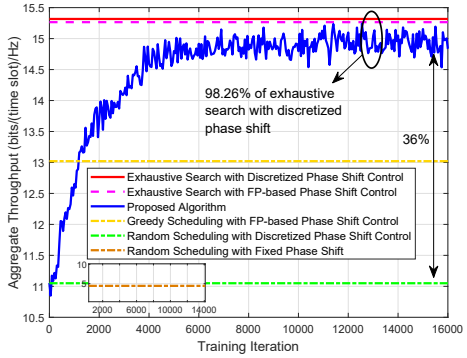

Fig. 4. Aggregate throughput versus the number of training iterations. We set $N = 20$, $M = 2$, and $L_R = 4$.

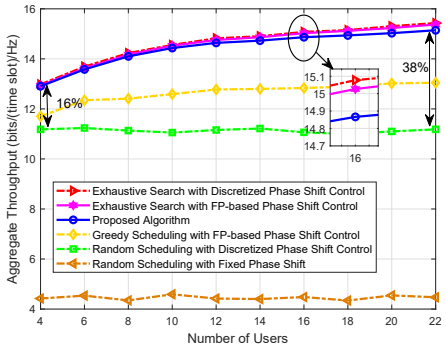

Fig. 5. Aggregate throughput versus the number of users. We set $M = 2$ and $L_R = 4$.

- Greedy user scheduling with FP-based phase-shift control: The base station schedules the M users with the largest $\left| h_{D,n}(t) + \boldsymbol{g}^H(t) \boldsymbol{h}_{R,n}(t) \right|^2$, $n \in \mathcal{N}$ and employs FP-based phase-shift control.
- Random scheduling with discretized phase-shift control: The base station randomly selects M users from the N users and applies discretized phase-shift control.
- Random scheduling with fixed phase-shift matrix: Apart from random scheduling, the phase shifts of all reflecting elements are set to 0.

Fig. 4 shows the evolution of the aggregate throughput versus training iterations. After 8000 training iterations (i.e., minibatches), the proposed NCO-based algorithm achieves an aggregate throughput that is 98.2% of the ES algorithm. The aggregate thoughput of the proposed algorithm is 36% higher than the random scheduling algorithm with discretized phase shift control. The results also show that, the performance of the proposed FP-based phase shift algorithm is comparable to the discretized phase shift control.

We vary the number of users and evaluate the performances in Fig. 5. The proposed algorithm is evaluated after 20000 training iterations. The results show that, the aggregate throughput of the proposed algorithm is 38% higher than that of the random scheduling with discretized phase shift control when the number of users $N = 22$. We observe that, IRS-aided systems benefits from properly scheduling the users and controlling the phase shift of the IRS.

Fig. 6 shows the impact of the number of reflecting elements L_R on the aggregate throughput. Only FP-based phase-shift control is performed due to the high computational

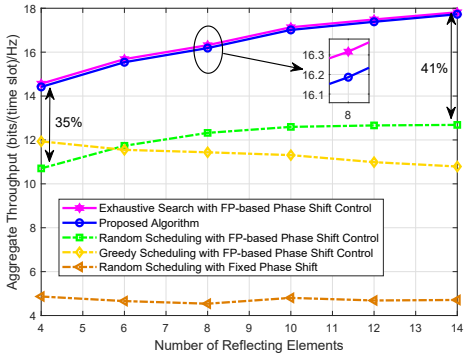

Fig. 6. Aggregate throughput versus the number of reflecting elements. We set $N = 10$ and $M = 2$.

complexity of the discretized phase-shift control under larger L_R. The proposed algorithm and random scheduling algorithm benefit from having more reflecting elements on the IRS. The greedy scheduling algorithm suffers performance degradation due to the increase in the interference as the received signal powers of the scheduled users may be relatively high.

For the computational complexity, we evaluate the runtime of different algorithms for 10 consecutive time slots on the same computing server. We set $N = 10$, $M = 2$, and $L_R = 4$. The runtime for the online execution of the proposed algorithm, the random scheduling, and the ES-based user scheduling with FP-based phase-shift control is 23.0, 22.1, and 999.7 sec, respectively. We observe that the computational complexity of the proposed algorithm is significantly lower than the ES-based algorithm and is close to the random scheduling algorithm.

V. Conclusion

In this paper, we proposed an NCO-based algorithm for the joint user scheduling and phase-shift control for throughput maximization in IRS-aided systems. We decomposed the formulated problem into two subproblems: a combinatorial optimization subproblem for user scheduling, and an FP subproblem for phase-shift optimization. We utilized NCO to learn a stochastic policy for determining the near-optimal user scheduling. We then used FP to solve the phase-shift optimization problem. Simulation results showed that the proposed algorithm can achieve an aggregate throughput that is close to the computational-expensive ES algorithm, and is higher than several heuristic algorithms. For future work, we will consider throughput optimization for IRS-aided systems where each base station is equipped with multiple antennas.

Appendix

We denote the set of the M scheduled users as $\mathcal{M}$. We drop the notation for time slot t here for simplicity. We define $\boldsymbol{\lambda} = (e^{-j\psi_1}, \ldots, e^{-j\psi_{L_R}}, \rho) \in \mathbb{C}^{L_R+1}$, where $\rho \in \mathbb{C}$ and $|\rho|^2 = 1$. We further define $\boldsymbol{\Lambda} = \boldsymbol{\lambda}\boldsymbol{\lambda}^H$ to replace the control variable. This leads to the following constraints [5]:

$$\mathrm{Diag}(\boldsymbol{\Lambda}) = \boldsymbol{I}_{L_R+1}, \tag{25}$$

$$\mathrm{rank}(\boldsymbol{\Lambda}) = 1, \tag{26}$$

where $\mathrm{Diag}(\boldsymbol{\Lambda})$ denotes the diagonal matrix whose diagonal elements are the same as that of $\boldsymbol{\Lambda}$, and $\boldsymbol{I}_{L_R+1}$ is an $(L_R+1)\times(L_R+1)$ identity matrix.

For user $m \in \mathcal{M}$, we define

$$\boldsymbol{\Theta}_m = \left[\left(\mathrm{diag}(\boldsymbol{h}_{R,m}^H)\boldsymbol{g}\right)^T \quad h_{D,m}^* \right]^T. \tag{27}$$

The SINR of user $m \in \mathcal{M}$ can now be rewritten as follows:

$$\Gamma_m = \frac{P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_m\boldsymbol{\Theta}_m^H)}{\sum_{j\in\mathcal{M}\setminus\{m\}} P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_j\boldsymbol{\Theta}_j^H)+\sigma^2}. \tag{28}$$

We use FP [10] to tackle the multi-ratio fractional objective function in subproblem (7). We apply the quadratic transform [10] and obtain the following problem

$$\begin{aligned} \underset{\boldsymbol{\Lambda}}{\text{maximize}} \quad & \sum_{m\in\mathcal{M}} \log_2\Big(1 + 2y_m\sqrt{P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_m\boldsymbol{\Theta}_m^H)} \\ & \qquad - y_m^2\Big(\sum_{j\in\mathcal{M}\setminus\{m\}} P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_j\boldsymbol{\Theta}_j^H)+\sigma^2\Big)\Big) \\ \text{subject to} \quad & \text{constraints (25), (26),} \end{aligned} \tag{29}$$

where y_m is resulted from the quadratic transform for each scheduled user. The optimal y_m for fixed $\boldsymbol{\Lambda}$ is given by

$$y_m^\star = \frac{\sqrt{P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_m\boldsymbol{\Theta}_m^H)}}{\sum_{j\in\mathcal{M}\setminus\{m\}} P^{\max}\,\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Theta}_j\boldsymbol{\Theta}_j^H)+\sigma^2}, \ m \in \mathcal{M}.$$

For fixed y_m, $m \in \mathcal{M}$, we use semidefinite relaxation to tackle constraint (26) and problem (29) is now a convex optimization problem that can be solved using standard solvers. A suboptimal solution of problem (29) can be obtained by iteratively optimizing y_m, $m \in \mathcal{M}$ and $\boldsymbol{\Lambda}$ [10].

References

[1] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive beamforming design," in *Proc. of IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018.

[2] X. Yu, D. Xu, and R. Schober, "MISO wireless communication systems via intelligent reflecting surfaces: (invited paper)," in *Proc. of IEEE/CIC Int'l Conf. Commun. in China (ICCC)*, Chengdu, China, Aug. 2019.

[3] S. Zhang and R. Zhang, "Intelligent reflecting surface aided multiple access: Capacity region and deployment strategy," in *Proc. of IEEE Int'l Workshop Signal Process. Advances Wireless Commun.*, May 2020.

[4] Y. Jia, C. Ye, and Y. Cui, "Analysis and optimization of an intelligent reflecting surface-assisted system with interference," in *Proc. of IEEE Int'l Conf. Commun. (ICC)*, Jun. 2020.

[5] D. Xu, X. Yu, and R. Schober, "Resource allocation for intelligent reflecting surface-assisted cognitive radio networks," in *Proc. of IEEE Int'l Workshop Signal Process. Advances Wireless Commun.*, May 2020.

[6] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *CoRR*, vol. abs/1611.09940, Jan. 2017.

[7] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. of Int'l Conf. Learn. Representations (ICLR)*, New Orleans, LA, May 2019.

[8] M. Ma and V. W. S. Wong, "Joint user pairing and association for multicell NOMA: A pointer network-based approach," in *Proc. of IEEE Int'l Conf. Commun. (ICC) Workshop*, Jun. 2020.

[9] A. Vaswani *et al.*, "Attention is all you need," in *Proc. of Advances in Neural Inf. Process. Sys. (NeurIPS)*, Long Beach, CA, Dec. 2017.

[10] K. Shen and W. Yu, "Fractional programming for communication systems - Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.

[11] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, May 1992.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of Int'l Conf. Learning Representations (ICLR)*, San Diego, CA, May 2015.

[13] 3GPP TR 38.901 V16.1.0, "Technical specification group radio access network; Study on channel model for frequencies from 0.5 to 100 GHz (Release 16)," Jan. 2020.