DRL-Based Sequential Scheduling for IRS-Assisted MIMO Communications

Dariel Pereira-Ruisánchez, *Student Member, IEEE*, Óscar Fresnedo, *Member, IEEE*, Darian Pérez-Adán, *Member, IEEE*, and Luis Castedo, *Senior Member, IEEE*

***Abstract*—Efficient resource allocation strategies are pivotal in vehicular communications as connected devices steeply increase in scenarios with much more stringent requirements. In this work, we propose a deep reinforcement learning (DRL)-based sequential scheduling approach for sum-rate maximization in the uplink of intelligent reflecting surface (IRS)-assisted multi-user (MU) multiple-input multiple-output (MIMO) vehicular communications. We formulate the scheduling task as a partially observable Markov decision process (POMDP) and propose a novel stream-level sequential solution based on the proximal policy optimization (PPO) algorithm. We consider a realistic imperfect channel state information (ICSI) model and assess the proposal in several communication setups comprising both spatially uncorrelated and correlated links. Simulation results show that the proposed DRL-based sequential scheduling approach is a robust alternative to more computationally demanding benchmarks.**

***Index Terms*—Scheduling, intelligent reflecting surfaces, deep reinforcement learning, PPO, resource allocation.**

I. Introduction

OVER the last few years, vehicle-related technologies have evolved to support advanced applications related to driving assistance and collision avoidance tasks. The high mobility and critical conditions under which these applications must work impose latency, reliability, and throughput requirements that cannot be achieved with current wireless technologies [1], [2].

IRS-assisted MIMO systems have been regarded as enablers of the next generation of vehicle-to-everything (V2X) communications. These technologies set the basis for providing ubiquitous coverage while supporting high-throughput, ultra-reliable and low-latency transmissions. On the one hand, MIMO systems achieve significant spatial multiplexing and enable more efficient spectrum usage, higher data rates, and privacy [3]. In addition, the use of IRSs allows to smartly control the communication environment and reduce signal degradation in high-frequency bands, which include millimeter wave (mmWave) and terahertz (THz) [4], [5]. The deployment of unmanned aerial vehicle (UAV)-carried IRSs is an attractive solution to enhance ground communications by creating artificial links between vehicles and roadside units (RSUs) that would be obstructed otherwise [6], [7].

Although the deployment of IRS-assisted MIMO systems might be a game-changing paradigm, several deployment considerations remain open problems. Recently, solving the joint optimization of the precoders and the IRS phase-shift matrix has attracted the research interest the most. However, results in [8] show that the performance of some promising solutions significantly degrades when the number of transmitted streams increases beyond the number of receiving antennas. As explained in [9] and [10], the next generations of vehicular communications will face highly heterogeneous and dynamic scenarios where loads of users, sensors, and vehicles will compete for the available resources. Hence, satisfactory communication performance will be unfeasible without addressing appropriate resource allocation techniques.

As discussed in [11], the selection of the co-scheduled users significantly affects performance in non-orthogonal transmissions like those in IRS-assisted MIMO systems. However, selecting the set of co-scheduled users that optimizes the system performance is a non-deterministic polynomial time problem because the set of feasible solutions grows exponentially with the number of users. As a result, many existing scheduling approaches consider sub-optimal heuristics to reduce the search over the high-dimensional solution space [9], [12]. These approaches use surrogate objective functions based on spatial compatibility metrics and perform the user scheduling sequentially. However, they only consider the immediate effect of incorporating a given user and disregard long-term effects over the final result. In addition, some of the spatial metrics considered cannot be extended to all the scenarios since they depend on channel properties like the channel correlation, which provides no useful information in uncorrelated channel models [9], [12].

Data-driven scheduling approaches have recently gained attention due to the ability of artificial neural networks (ANNs) to solve high-dimensional resource allocation problems in an efficient and flexible way [13], [14]. In this regard, DRL approaches—which combine reinforcement learning (RL) with ANN-based function approximations—stand as the most appealing alternatives [10]. DRL frameworks learn how to solve

Manuscript received 26 September 2023; revised 2 January 2024; accepted 24 January 2024. Date of publication 26 January 2024; date of current version 20 June 2024. This work was supported by MCIN/AEI/10.13039/501100011033 under Grants PID2019-104958RB-C42 (ADELE) and PID2022-137099NB-C42 (MADDIE), in part by Marie Sklodowska-Curie through European Union's Horizon 2020 Research and Innovation Programme under Grant 101034261, and in part by the Consellería de Cultura, Educación e Universidade of the Xunta de Galicia. The review of this article was coordinated by Dr. Shaowei Wang. *(Corresponding author: Dariel Pereira-Ruisánchez.)*

The authors are with the Department of Computer Engineering & CITIC Research Center, University of A Coruña, 15001 A Coruña, Spain (e-mail: d.ruisanchez@udc.es; oscar.fresnedo@udc.es; d.adan@udc.es; luis.castedo@udc.es).

Digital Object Identifier 10.1109/TVT.2024.3359117

optimization problems by continuously interacting with the environment, becoming a robust alternative for rapidly time-varying channels. Authors in [11] use a dueling double deep Q-network (D3QN) formulation to find the user association strategies that maximize the long-term downlink performance of a cellular network. The scheduling problem is addressed as a tree-structured combinatorial problem, and an adaptation of the deep Q-network (DQN) framework is employed. In [15], a multi-agent formulation based on DQN is considered to address the resource allocation in UAV-enabled communications. The results in [11] and [15] show that DRL-based approaches outperform conventional alternatives while offering a better trade-off between execution time and adaptability. In addition, DRL approaches in [11] and [15] leverage the attention to long-term rewards in order to achieve near-optimal solutions. However, these implementations suffer from *the curse of dimensionality* [16], [17] since they consider combinatorial approaches whose set of actions comprises all the feasible scheduling combinations. As a result, the set of actions increases exponentially with the number of users, thus leading to unfeasible storage and computing requirements.

Because of the limitations of existing data-driven schedulers, and the inability of heuristic methods to perform long-term analysis, we propose an innovative approach that combines the best of the sequential formulations and DRL algorithms to handle the user scheduling in the uplink of IRS-assisted multi-stream (MS) MU MIMO communications. We developed an efficient and robust scheduling framework such that practical strategies for the joint optimization of the IRS matrix and MIMO precoders could be later derived from its output.

The remainder of this paper is structured as follows. Section II details some theoretical fundamentals of RL, analyzes the most relevant existing DRL-based scheduling approaches, and presents the main contributions of our work. Section III introduces the IRS-assisted MS MU MIMO system model and the scheduling optimization problem. Section IV introduces the proposed sequential DRL-based solution. Section V presents the results of simulation experiments, and Section VI is devoted to the conclusions.

A. Notation

Along this work, the following notation will be employed: a is a scalar, $\mathbf{a}$ is a column vector, and $\mathbf{A}$ represents a matrix. Notice that we use the scalar notation for actions and states in Section II-A, but their formats vary according to the RL formulation of the problems. $[\mathbf{A}]_{i,:}$ and $[\mathbf{A}]_{:,j}$ stand for the i-th row vector and the j-th column vector of $\mathbf{A}$, respectively. $[\mathbf{A}]_{i,j}$ is the entry where the i-th row and the j-th column of $\mathbf{A}$ meet. Transpose, conjugate transpose, and the Frobenius norm of $\mathbf{A}$ are represented by $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}^{\mathrm{H}}$, and $\|\mathbf{A}\|_{\mathrm{F}}^2$, respectively. $\hat{\mathbf{A}}$ represents the estimate of the matrix $\mathbf{A}$ and $\widehat{\mathbf{AB}}$ stands for the estimate of the result of the matrix operation between $\mathbf{A}$ and $\mathbf{B}$. Calligraphic letters are employed to denote sets and tuples. $|\mathcal{R}|$ stands for the cardinality of a set $\mathcal{R}$. $\mathbf{I}_N$ indicates an $N \times N$ identity matrix, and $\mathcal{I}_N$ denotes the set of integers from 1 to N. We use $\mathbf{0}$ to represent indistinctly zero-valued vectors or matrices whose dimensions can be easily inferred. The operator $\mathrm{blkdiag}(\cdot)$ constructs a block diagonal matrix from its input matrices, the operator $\mathrm{diag}(\cdot)$ constructs a diagonal matrix from an input vector, and $\mathrm{flatten}(\cdot)$ is the operator that reshapes any matrix $\mathbf{V} \in \mathbb{C}^{A\times B}$ into a vector $\mathbf{v} \in \mathbb{C}^{AB}$. Finally, $\mathrm{OR}(\cdot)$ computes the element-wise binary OR operation between the binary-valued input vectors. The mathematical relationships presented in the following sections hold for all the consecutive time steps t that fit within one coherence block. Hence, for the sake of simplicity, sub-index t is used only where necessary to avoid ambiguities.

II. Motivation and Contributions

A. Reinforcement Learning (RL) Fundamentals

As stated in [16], RL is a computational approach to learning-by-interacting, i.e., mapping situations to the actions that maximize a numerical reward function. Most RL problems are formalized in terms of a Markov decision process (MDP), where the learning and decision-maker element (the agent) interacts with the external components (the environment) through actions that affect the subsequent states and rewards. Hence, RL problems are characterized by a dynamics function $p(s_{t+1}, r_t|a_t, s_t)$ such that, in every time instant t, the next state s_{t+1}, and the reward r_t are conditioned by the effect of taking an action $a_t \in \mathcal{A}$ in the current state s_t. $\mathcal{A}$ is the set of feasible actions and $|\mathcal{A}|$ is the number of feasible actions.

The policy $\pi(\cdot)$ is a critical element of RL approaches. The policy is the decision-making rule that returns the probability $\pi(a_t|s_t)$ for taking a given action a_t while being in a given state s_t. RL algorithms aim to maximize a function of the expected long-term reward, which depends on the sequence of states and actions taken. Hence, training in RL focuses on learning the policy that maximizes that reward function. The state-value function $V_\pi(s_t)$ and the action-value function $Q_\pi(s_t, s_t)$ are the most commonly used reward functions. The former computes the expected return when starting in a state s_t and following the policy $\pi(\cdot)$, while the latter evaluates the expected return starting from s_t, taking the action a_t and following policy $\pi(\cdot)$ afterward. The tuples $\mathcal{E}_t = (s_t, a_t, r_t, s_{t+1}), \forall t$ that store the interactions between the RL elements are commonly termed experiences. The structure of the experience tuples may change according to the RL algorithm.

Conventional tabular approaches to RL problems have proven efficient when considering low-dimensional and discrete state and action spaces [16]. However, these algorithms are unfeasible in optimization problems with continuous or arbitrarily large search spaces. In this regard, the ANN-based DRL algorithms are appealing alternatives. The use of ANNs for the approximations of the policy and reward functions enables a wide range of new approaches to high-complexity problems like the one we are addressing in this work. In this case, the objective function is a determining factor since the trainable parameters of the ANNs update according to it. Hence, several recent advances in DRL have been motivated by the search for better objective functions.

B. Related Works

We next analyze three existing approaches to user scheduling in IRS-assisted communications where DRL-based techniques

have been considered. Although these works also address the configuration of the IRS matrix, we focus only on the scheduling stage, which is the scope of this work.

Authors in the complementary works [18] and [19] address the user scheduling task in the uplink of an IRS-assisted communication system. They consider a sum-rate maximization problem and propose a solution based on the neural combinatorial optimization (NCO) framework. NCO is a stochastic method widely used to handle RL formulations where the solutions comprise the combinations of the optimization variables. This framework overcomes the high dimensionality of combinatorial problems by considering a sequential recurrent structure. The simulation results show that the proposed algorithms achieve near-optimal performances in the considered scenarios. However, the authors in [18] and [19] make some questionable assumptions. First, they assume a fixed number of scheduled users along all the channel realizations, although the optimal number of scheduled users varies according to the characteristics of the communication channels. Second, when analyzing the impact of the imperfect CSI (ICSI), they use ICSI models where the channel matrices (from the users to the IRS and from the IRS to the base station (BS)) are individually estimated. However, because of the passive nature of the IRSs, these estimated matrices cannot be obtained separately in practical situations [20].

Authors in [21] propose a solution based on the PPO framework to the scheduling problem in the downlink of IRS-assisted vehicular communications. Unlike [18] and [19], the authors in [21] aim to maximize the minimum average rate experienced by the vehicles. The PPO algorithm is simpler to implement and tune, and the results in this paper show that it enables an efficient approach for user scheduling in the considered simulation scenarios. However, this approach cannot be extended to denser configurations. The major drawback of [21] is related to the formulation of the actions and the scalability of the resulting set. The authors in [21] propose a combinatorial approach where the feasible actions comprise all user scheduling combinations. Hence, considering denser networks with more competing vehicles might lead to intractable computing requirements.

Notice that all previous related works consider single-antenna users and single-stream transmissions. This simplification limits the scheduling capabilities to the user level and disregards the advantages of considering stream-level scheduling. Besides, they only assess uncorrelated fading channels while, as explained in [22], practical channels are generally spatially correlated. Because of the antenna's non-uniform radiation patterns and the physical propagation environment, some spatial directions are more likely to carry strong signals.

C. Overcoming the Limitations

Based on the limitations of the previous related works, we have developed the present research, whose main contributions can be summarized as follows:

- We propose a sequential DRL-based scheduling framework for IRS-assisted MIMO communications. This proposal leverages the attention to long-term rewards in RL algorithms and remains feasible for highly populated communication systems.
- We formulate the scheduling optimization problem as a partially observable MDP (POMDP) where observations are composed of the estimated channel state information (CSI) matrices, and actions are related to selecting the user at each step of the sequential scheduler. We develop a DRL solution termed PPO, which efficiently handles the high-dimensional continuous states and discrete actions.
- We extend the scheduling optimization to the stream level, which provides higher flexibility and enhanced performance regarding conventional user-level scheduling approaches.
- We consider a realistic ICSI model where the individual channel matrices in the cascaded channels are jointly estimated. Besides, we evaluate performance over several communication scenarios, including both correlated and uncorrelated channels.

In this work, we have selected a PPO-based approach as it stands as a game-changing framework regarding stability, simplicity, and scalability in DRL algorithms. As we will explain later, PPO is an innovative policy gradient algorithm that introduces several improvements to overcome major constraints like sample inefficiency and the instability of policy updates. During the initial phases of the investigation, we also considered other DRL-based algorithms. However, those lacked some desired features (e.g., DQN and advantage actor-critic (A2C)) or were too complex to be considered for practical implementations (e.g., trust region policy optimization (TRPO)).

In addition, we formalize the scheduling problem as a POMDP because the interactions between the RL elements are always affected by the uncertainty introduced by the considered ICSI conditions. Although we will not distinguish between the terms states and observations, we assume that the observed rewards are affected by a non-observable element: the CSI estimation errors.

III. System Model and Optimization Problem

Let us consider the uplink of an IRS-assisted MS MU MIMO vehicular communication system, where the communication between K vehicles and an RSU is aided by a UAV-carried IRS, as shown in Fig. 1. The set of all the connected vehicles is $\mathcal{K} = \{k : k = 1, \ldots, K\}$. We assume each vehicle uses N_{t} antennas to send up to N_{s} data streams to the RSU equipped with N_{r} antennas. We consider the vehicles to transit in a dark zone, i.e., there are no direct paths between the vehicles and the RSU. Hence, coverage is provided through the vehicle-IRS-RSU cascaded channels. According to this system model, the signal received at the RSU is given by

$$\mathbf{y} = \mathbf{H}_{\mathrm{IR}}\boldsymbol{\Theta}\mathbf{H}_{\mathrm{VI}}\mathbf{P}\boldsymbol{\Xi}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x} = [\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_K^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{KN_{\mathrm{s}}}$ stacks the symbols transmitted by the K vehicles, with $\mathbf{x}_k \in \mathbb{C}^{N_{\mathrm{s}}}$, $\forall k$. Vector $\mathbf{n} \in \mathbb{C}^{N_{\mathrm{r}}}$ stands for the receive complex-valued additive white Gaussian noise (AWGN), modeled as $\mathbf{n} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \sigma^2\mathbf{I}_{N_{\mathrm{r}}})$. We assume

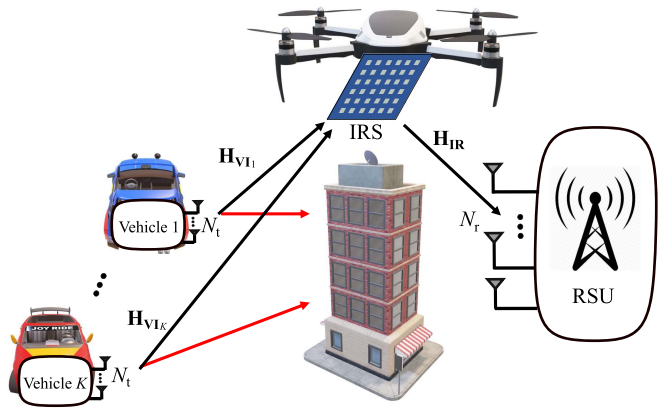


Fig. 1. Uplink of an IRS-assisted MIMO communication between several connected vehicles and an RSU.

every $\mathbf{x}_k$ follows a circularly symmetric complex Gaussian distribution, i.e., $\mathbf{x}_k \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I}_{N_{\mathrm{s}}})$.

The diagonal matrix $\boldsymbol{\Xi} = \mathrm{diag}(\boldsymbol{\xi}) \in \mathbb{C}^{N_{\mathrm{s}}K \times N_{\mathrm{s}}K}$ is the stream-level scheduling matrix. The vector $\boldsymbol{\xi} = [\xi_{1,1}, \ldots, \xi_{K,N_{\mathrm{s}}}]^{\mathrm{T}}$ is a binary scheduling vector, such that $\xi_{k,n_{\mathrm{s}}}$ takes the value one if the n_{s}-th stream of the k-th vehicle is scheduled and zero otherwise. We define the set of scheduled vehicles as $\mathcal{S}_{\boldsymbol{\xi}} = \{s \in \mathcal{K} : \sum_{n_{\mathrm{s}}=1}^{N_{\mathrm{s}}} \xi_{s,n_{\mathrm{s}}} \neq 0\}$, such that it contains all the vehicles with at least one scheduled stream according to $\boldsymbol{\xi}$.

The matrix $\mathbf{P} = \mathrm{blkdiag}(\mathbf{P}_1, \ldots, \mathbf{P}_K) \in \mathbb{C}^{N_{\mathrm{t}}K \times N_{\mathrm{s}}K}$ stacks all the individual precoders $\mathbf{P}_k \in \mathbb{C}^{N_{\mathrm{t}} \times N_{\mathrm{s}}}$. We assume the IRS to have N reflecting elements. Hence, we model its phase-shift matrix as the diagonal matrix $\boldsymbol{\Theta} = \mathrm{diag}(\boldsymbol{\theta}) \in \mathbb{C}^{N \times N}$ where $\boldsymbol{\theta} = [e^{\mathrm{j}\theta_1}, \ldots, e^{\mathrm{j}\theta_N}]^{\mathrm{T}} \in \mathbb{C}^N$ is the vector that stacks the phase shifts introduced by the elements of the IRS.

The matrix $\mathbf{H}_{\mathrm{IR}} \in \mathbb{C}^{N_{\mathrm{r}} \times N}$ is the channel response from the IRS to the RSU, and $\mathbf{H}_{\mathrm{VI}} = [\mathbf{H}_{\mathrm{VI}1}, \ldots, \mathbf{H}_{\mathrm{VI}K}] \in \mathbb{C}^{N \times N_{\mathrm{t}}K}$ stacks the channel response matrices of the links between the vehicles and the IRS. Hence, the cascaded channel matrix $\mathbf{H}_{\mathrm{C}}$ can be defined as $\mathbf{H}_{\mathrm{C}} = \mathbf{H}_{\mathrm{IR}}\boldsymbol{\Theta}\mathbf{H}_{\mathrm{VI}} \in \mathbb{C}^{N_{\mathrm{r}} \times N_{\mathrm{t}}K}$, where $\mathbf{H}_{\mathrm{C}}$ stacks all the individual cascaded channels, i.e., $\mathbf{H}_{\mathrm{C}} = [\mathbf{H}_{\mathrm{C}_1}, \ldots, \mathbf{H}_{\mathrm{C}_K}]$ with $\mathbf{H}_{\mathrm{C}_k} = \mathbf{H}_{\mathrm{IR}}\boldsymbol{\Theta}\mathbf{H}_{\mathrm{VI}_k}, \forall k$. Note that the response of the cascaded channel can be rewritten as

$$\mathbf{H}_{\mathrm{C}} = \sum_{n=1}^{N} \mathbf{H}_{\mathrm{com}_n} e^{\mathrm{j}\theta_n}, \tag{2}$$

where $\mathbf{H}_{\mathrm{com}_n} = [\mathbf{H}_{\mathrm{IR}}]_{:,n}[\mathbf{H}_{\mathrm{VI}}]_{n,:} \in \mathbb{C}^{N_{\mathrm{r}} \times N_{\mathrm{t}}K}$ is the combined channel matrix response for the n-th element of the IRS.

We next introduce the ICSI model to be considered. As shown in [20], estimates of the channel matrices $\hat{\mathbf{H}}_{\mathrm{IR}}$ and $\hat{\mathbf{H}}_{\mathrm{VI}}$ cannot be obtained individually due to the passive nature of the IRS elements. Alternatively, estimates of the combined channel response $\hat{\mathbf{H}}_{\mathrm{com}_n} = \widehat{[\mathbf{H}_{\mathrm{IR}}]_{:,n}[\mathbf{H}_{\mathrm{VI}}]_{n,:}}, \forall n$ are feasible at the RSU. Such estimates can be modeled as [20]

$$\hat{\mathbf{H}}_{\mathrm{com}_n} = \underbrace{[\mathbf{H}_{\mathrm{IR}}]_{:,n}\,[\mathbf{H}_{\mathrm{VI}}]_{n,:}}_{\mathbf{H}_{\mathrm{com}_n}} + \mathbf{E}_{\mathrm{com}n}, \tag{3}$$

where $\mathbf{H}_{\mathrm{IR}}$ and $\mathbf{H}_{\mathrm{VI}}$ are the true channel matrices, and the matrices $\mathbf{E}_{\mathrm{com}n}, \forall n$ contain the estimation errors, which are assumed to be zero-mean Gaussian distributed with covariance matrices $\mathbb{E}[\mathbf{e}_{\mathrm{com}_n}\mathbf{e}_{\mathrm{com}_n}^{\mathrm{H}}] = \sigma_{\mathrm{com}_n}^2 \mathbf{I}_{KN_{\mathrm{t}}N_{\mathrm{r}}}, \forall n$, where $\mathbf{e}_{\mathrm{com}_n}$ is the vectorized version of the n-th estimation error matrix. Hence, the estimated cascaded channel matrix can be represented as

$$\hat{\mathbf{H}}_{\mathrm{C}} = \sum_{n=1}^{N} \underbrace{\left([\mathbf{H}_{\mathrm{IR}}]_{:,n}\,[\mathbf{H}_{\mathrm{VI}}]_{n,:} + \mathbf{E}_{\mathrm{com}n}\right)}_{\hat{\mathbf{H}}_{\mathrm{com}_n}} e^{\mathrm{j}\theta_n}. \tag{4}$$

Notice that $\hat{\mathbf{H}}_{\mathrm{C}}$ stacks all the individual estimated cascaded channels such that $\hat{\mathbf{H}}_{\mathrm{C}} = [\hat{\mathbf{H}}_{\mathrm{C}_1}, \ldots, \hat{\mathbf{H}}_{\mathrm{C}_K}]$.

Next, the vector of estimated symbols at the RSU $\hat{\mathbf{x}}$ can be obtained by linear filtering the received signal, i.e., $\hat{\mathbf{x}} = \mathbf{W}^{\mathrm{H}}\mathbf{y}$, being $\mathbf{W}^{\mathrm{H}} = [\mathbf{W}_1, \ldots, \mathbf{W}_K]^{\mathrm{H}} \in \mathbb{C}^{KN_{\mathrm{s}} \times N_{\mathrm{r}}}$ the RSU receiving filter matrix that stacks all the individual receiving filter matrices $\mathbf{W}_k^{\mathrm{H}} \in \mathbb{C}^{N_{\mathrm{s}} \times N_{\mathrm{r}}}$. We assume $\mathbf{W}_k^{\mathrm{H}}, \forall k$ to be the minimum mean square error (MMSE) filters, which are computed as in [23] and [22] by

$$\mathbf{W}_k^{\mathrm{H}} = \mathbf{P}_k^{\mathrm{H}}\hat{\mathbf{H}}_{\mathrm{C}_k}^{\mathrm{H}} \left(\hat{\mathbf{H}}_{\mathrm{C}}\mathbf{P}\mathbf{P}^{\mathrm{H}}\hat{\mathbf{H}}_{\mathrm{C}}^{\mathrm{H}} + \sigma_{\mathrm{n}}^2\mathbf{I}_{N_{\mathrm{r}}}\right)^{-1}. \tag{5}$$

Now, we can formulate the optimization problem to determine the scheduling vector that maximizes the system sum-rate ($\Lambda_{\boldsymbol{\xi}}$) as follows

$$\arg\max_{\boldsymbol{\xi}} \Lambda_{\boldsymbol{\xi}}, \tag{6}$$

with

$$\Lambda_{\boldsymbol{\xi}} = \sum_{s \in \mathcal{S}_{\boldsymbol{\xi}}} R_s,$$

where R_s is the individual rate of the s-th vehicle given by

$$R_s = \log_2 \det\left(\mathbf{I}_{N_{\mathrm{s}}} + \mathbf{X}_s^{-1}\mathbf{W}_s^{\mathrm{H}}\mathbf{H}_{\mathrm{C}_s}\mathbf{P}_s\mathbf{P}_s^{\mathrm{H}}\mathbf{H}_{\mathrm{C}_s}^{\mathrm{H}}\mathbf{W}_s\right), \tag{7}$$

where

$$\mathbf{X}_s = \sum_{i \neq s} \mathbf{W}_s^{\mathrm{H}}\mathbf{H}_{\mathrm{C}_i}\mathbf{P}_i\mathbf{P}_i^{\mathrm{H}}\mathbf{H}_{\mathrm{C}_i}^{\mathrm{H}}\mathbf{W}_s + \sigma^2\mathbf{W}_s^{\mathrm{H}}\mathbf{W}_s \tag{8}$$

is the interference plus noise matrix. For greater clarity, Table I summarizes the main system model parameters and their descriptions.

Let us recall that we must use the estimated channel matrices for computing the entries of $\boldsymbol{\xi}, \boldsymbol{\Theta}, \mathbf{P}$, and $\mathbf{W}^{\mathrm{H}}$. However, we will use the true channel matrices when determining the system sum-rate values for a realistic analysis of the system performance.

Some conventional approaches require computing the matrices $\boldsymbol{\Theta}$, $\mathbf{P}$, and $\mathbf{W}^{\mathrm{H}}$ in every scheduling step. They also require executing an alternating optimization algorithm that iterates between the scheduling and system optimization stages. However, the proposed DRL-based framework enables us to fully separate the scheduling optimization task and determine these matrices only for the final scheduling vector $\boldsymbol{\xi}$. As we explain later, our proposal entails no knowledge of the IRS and precoder matrices along the steps in the online scheduling algorithm. Since matrices $\mathbf{P}$ and $\mathbf{W}^{\mathrm{H}}$ are computed for a given scheduling vector, we can assume that columns/rows related to

TABLE I
SYSTEM MODEL PARAMETERS

Parameter	Description
K	number of connected vehicles
$\mathcal{K}$	set of connected vehicles
N_{t}	number of transmitting antennas per vehicle
N_{s}	number of transmitted streams per vehicle
N_{r}	number of receiving antennas at the RSU
N	number of reflecting elements at the IRS
$\mathbf{H}_{\mathrm{VI}}$	true channel matrix from the vehicles to the IRS
$\mathbf{H}_{\mathrm{IR}}$	true channel matrix from the IRS to the RSU
$\mathbf{H}_{\mathrm{C}}$	true cascaded channel matrix
$\hat{\mathbf{H}}_{\mathrm{C}}$	estimated cascaded channel matrix
$\mathbf{H}_{\mathrm{com}_n}$	true combined channel matrices
$\hat{\mathbf{H}}_{\mathrm{com}_n}$	estimated combined channel matrices
$\mathbf{E}_{\mathrm{com}\,n}$	estimation error matrices
$\mathbf{P}$	block-diagonal matrix with all the users' precoders
$\boldsymbol{\Theta}$	IRS phase-shift matrix
$\boldsymbol{\theta}$	IRS phase-shift vector
$\boldsymbol{\Xi}$	scheduling matrix
$\boldsymbol{\xi}$	scheduling vector
$\mathcal{S}_{\boldsymbol{\xi}}$	set of scheduled vehicles
$\mathbf{W}^{\mathrm{H}}$	MMSE receiving filter matrix
R_s	achievable rate for scheduled vehicle s
$\Lambda_{\boldsymbol{\xi}}$	achievable system sum-rate

the non-scheduled streams take zero-valued entries and do not impact the sum-rate.

During the training of the PPO-based scheduler and subsequent performance analysis, we compute the IRS matrix $\boldsymbol{\Theta}$ and the precoders' matrix $\mathbf{P}$ using the DCB-DDPG framework introduced in [8]. The simulation results in [8] show that this framework achieves near-optimal performance in several communication scenarios, even under ICSI conditions, which allows us to focus on the scheduling framework. However, note that the proposed PPO-based scheduler is independent of the implementation used for IRS and precoder optimization, so other alternatives could also be considered.

As stated in [8], the computed precoders meet individual power constraints $\| \mathbf{P}_k \|_{\mathrm{F}}^2 \leq \Omega_k, \forall k$, where Ω_k represents the available power at the k-th vehicle. For simplicity, we assume the same power constraint value for all vehicles, i.e., $\Omega_k = \Omega,\ \forall k$. Without loss of generality, we also set the noise variance σ^2 equal to one. Therefore, the signal-to-noise ratio (SNR) in dB per user is given by $\mathrm{SNR} = 10\log_{10}(\Omega)$.

In this work, two types of scenarios will be considered depending on whether a spatial correlation exists between the channel responses of the different vehicles or not. In the following subsections, we describe the channel models for these two types of scenarios.

A. Spatially Uncorrelated Channel Modeling

When considering spatially uncorrelated setups, we assume the links between the vehicles and the IRS follow an uncorrelated Rayleigh fading model: the entries of $\mathbf{H}_{\mathrm{VI}k}, \forall k$ are independent and identically distributed (i.i.d.) random variables such that $\mathbf{H}_{\mathrm{VI}k} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \beta_k \mathbf{I}_N)$, where β_k is the average channel gain for the vehicle k.

We also assume that the IRS is installed so that a line-of-sight (LoS) to the RSU exists. Therefore, a Rician fading channel

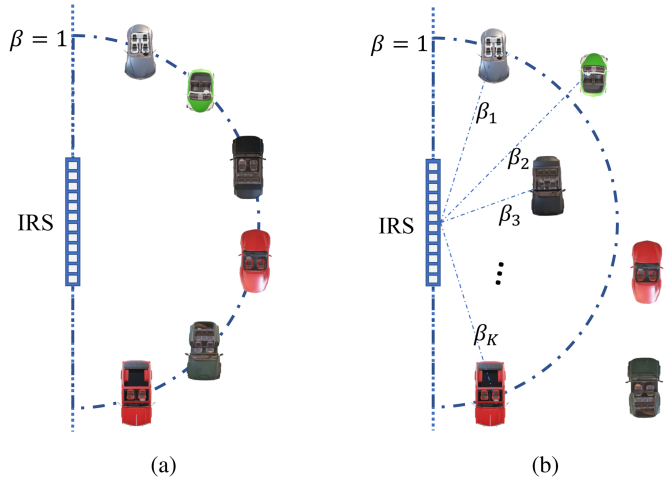


Fig. 2. Uncorrelated channel scenarios. (a) Constant distance. (b) Varying distance.

model is adopted to describe $\mathbf{H}_{\mathrm{IR}}$. As in [24], [25], [26], $\mathbf{H}_{\mathrm{IR}}$ is given by

$$\mathbf{H}_{\mathrm{IR}} = \sqrt{\frac{\psi}{1+\psi}}\mathbf{H}_{\mathrm{IR}}^{\mathrm{LOS}} + \sqrt{\frac{1}{1+\psi}}\mathbf{H}_{\mathrm{IR}}^{\mathrm{NLOS}}, \tag{9}$$

where ψ is the Rician factor, which is set to $\psi = 3$. $\mathbf{H}_{\mathrm{IR}}^{\mathrm{LOS}}$ and $\mathbf{H}_{\mathrm{IR}}^{\mathrm{NLOS}} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I}_N)$ are the LoS component and the non-line-of-sight (NLoS) uncorrelated Rayleigh fading component, respectively. For simplicity, we assume the RSU is equipped with a uniform linear array (ULA), and $\mathbf{H}_{\mathrm{IR}}^{\mathrm{LOS}}$ is computed as in [22].

At the same time, two different spatially uncorrelated situations will be considered for the vehicle-IRS links as shown in Fig. 2. The one on the left side of Fig. 2 (constant distance) assumes the average channel gains $\beta_k = 1, \forall k$. As in [22], we consider β_k stands for macroscopic large-scale fading related to distance-dependent path loss. Hence, this condition represents vehicles located the same distance from the IRS. On the right side, distances to the IRS are randomly distributed (varying distance). In this configuration, the average path gains follow $\beta_k \sim \mathcal{N}_{\mathbb{R}}(1,1),\ \forall k$. In practice, we use $|\beta_k|$ to avoid negative values.

B. Spatially Correlated Channel Modeling

In the second type of scenario, we assume the IRS reflecting elements are spatially correlated. The channels between the vehicles and the IRS are now described as $\mathbf{H}_{\mathrm{VI}k} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \beta_k \mathbf{R}_k), \forall k$, where $\mathbf{R}_k \in \mathbb{C}^{N \times N}$ is the positive semi-definite spatial correlation matrix for the k-th user. On the other hand, the correlated Rayleigh fading component of the channel between the IRS and the RSU is given by $\mathbf{H}_{\mathrm{IR}}^{\mathrm{NLOS}} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R}_{\mathrm{IR}})$.

As in [21], we assume the IRS is equipped with a ULA. Hence, the entries of the spatial correlation matrices are computed using the following simplified equation [22]:

$$[\mathbf{R}]_{a,b} = e^{2\pi \mathrm{j} d_{\mathrm{H}}(a-b)\sin(\phi)} e^{-\frac{\sigma_\phi^2}{2}(2\pi d_{\mathrm{H}}(a-b)\cos(\phi))^2}, \tag{10}$$

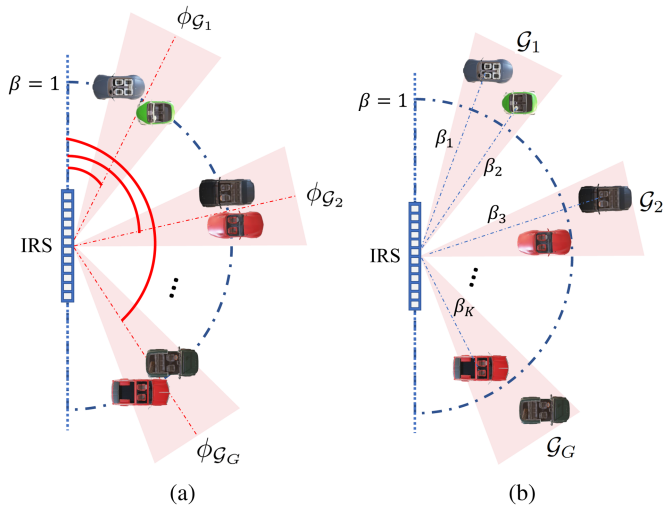


Fig. 3. Correlated channel scenarios. (a) Constant distance. (b) Varying distance.

where ϕ is the nominal angle in radians, σ_ϕ is the angular standard deviation in radians, and d_{H} is the antenna spacing measured in multiples of the wavelength. Equation (10) holds for computing the entries of the spatial correlation matrices $\mathbf{R}_k, \forall k$, and $\mathbf{R}_{\mathrm{IR}}$, by considering the nominal angles $\phi_k, \forall k$, and ϕ_{IR}, respectively. We assume $\sigma_\phi = 0.17$ rad $(\approx 10^\circ)$ for all the correlation matrices since it is a reasonable value in urban cellular networks [22]. Besides, we set the antenna spacing $d_{\mathrm{H}} = 0.5$, which is a commonly used value [22].

Fig. 3 illustrates the two spatially correlated situations considered for the vehicle-IRS links. In both configurations, vehicles are gathered in G sets $(\mathcal{G}_g, \forall g)$ with nominal angles $\phi_{\mathcal{G}_g}$ such that $\phi_k \sim \mathcal{N}_{\mathbb{R}}(\phi_{\mathcal{G}_g}, 0.05),\ \forall k \in \mathcal{G}_g$. Hence, vehicles in the same set have similar nominal angles and, thus, similar correlation matrices. Forcing this condition makes scheduling more challenging since vehicles with similar spatial properties strongly interfere between them. As in the spatially uncorrelated case, there is a configuration where the vehicles are located at the same distance (correlated constant distance) and another where the distances are different (correlated varying distance).

IV. PPO-Based Sequential Scheduling

This section describes the proposed sequential scheduling based on the PPO framework. First, some key features of this DRL-based framework are reviewed. Next, the state, action, and reward spaces considered for the optimization problem are defined. Finally, the algorithmic solution is derived.

A. PPO Framework

PPO is a model-free policy-gradient framework that was first introduced in [27] and has attracted a great deal of research interest. As explained in [27] and [28], PPO achieves a balanced performance. It combines the sample efficiency and policy improvement reliability of algorithms like TRPO while performing computationally tractable policy updates. Some of the features of a PPO agent which make it an appropriate choice for solving our optimization problem are the following:

- Actor-critic: called this way due to the interactions between the ANN-based approximations of the reward and policy functions. The critic, $\nu : (\mathbf{s}_t, \boldsymbol{\vartheta}_{\mathrm{v}}) \mapsto \hat{V}(\mathbf{s}_t)$, is the framework element that learns to map the state into an approximate of the state-value function. On the other hand, the actor, $\varpi : (\mathbf{s}_t, \boldsymbol{\vartheta}_\pi) \mapsto \mathbf{a}_t$, learns actions that maximize the critic's output. The trainable parameters of the critic, $\boldsymbol{\vartheta}_{\mathrm{v}}$, and the actor, $\boldsymbol{\vartheta}_\pi$, are updated by performing stochastic gradient descent updates over a joint objective function. As explained in [28] and [29], using a critic function approximation provides better stability since the variance of reward values used for training decreases. This reduction may lead to a faster convergence of the actor and critic networks than other actor-only policy gradient algorithms.
- Stochastic actions: the actor in PPO is composed of three stages. The first is the ANN-based approximation of the policy and computes a probability mass function for the feasible actions, i.e., $\pi(\mathbf{a}|\mathbf{s}_t, \boldsymbol{\vartheta}_\pi),\ \forall \mathbf{a} \in \mathcal{A}$. The second stage defines a discrete probability distribution from the computed probability mass function. Finally, a random sampler selects an action $(\mathbf{a}_t)$ according to the defined probability distribution. This stochastic selection of the discrete actions has natural capabilities for balancing exploration and exploitation during the training phase. Besides, stochastic actions are more desirable in POMDP since the probabilistic selection of actions prevents policies from getting stuck in wrong actions provoked by erroneous state observations.
- Multi-environment on-policy training: on-policy algorithms like PPO only use training experiences generated with the current policy. To enhance the efficiency of the sample generation task, PPO performs a multi-environment process where several agents with the same current policy run in parallel over different environments. These experiences are stored in a replay buffer (which we will call $\mathcal{R}$), which refills with the up-to-date collected data at every training stage.
- Clipped objective function: this is the most relevant feature of PPO algorithms. The objective function used by PPO to train the policy network avoids destructively large updates by clipping the output of one of the elements of the objective function. As updates are limited to not significantly altering the existing policy, multiple updates are possible over the same up-to-date collected data. Hence, it provides higher stability for training since it prevents the ANN-based policy from suffering the effects of vanishing or exploding gradients.

B. State, Action and Reward

In order to solve the scheduling optimization problem in (6), we introduce the following states, actions, and rewards. The **state** vector $\mathbf{s}_t$ comprises the entries of the current scheduling vector $\boldsymbol{\xi}_t$ and the entries of the estimated cascaded channel

matrix $\hat{\mathbf{H}}_{\mathrm{C}}$. Hence, the state vector is constructed such that

$$\mathbf{s}_t = [\underbrace{\xi_{1,1}, \ldots, \xi_{K,N_{\mathrm{s}}}}_{\boldsymbol{\xi}_t}, \mathrm{flatten}(\hat{\mathbf{H}}_{\mathrm{C}})]^{\mathrm{T}}. \tag{11}$$

The estimated cascaded channel matrix $\hat{\mathbf{H}}_{\mathrm{C}}$ is computed as in (4) by considering the estimated combined channel matrices $\hat{\mathbf{H}}_{\mathrm{com}_n}, \forall n$ and assuming an IRS matrix $\boldsymbol{\Theta} = \mathbf{I}_N$ along all the training steps. This way, the information in the state vectors is only affected by variations in the estimated combined channel matrices. We consider each training episode to fit within one channel coherence block. Hence, $\hat{\mathbf{H}}_{\mathrm{C}}$ does not vary during each training episode.

For every episode, the initial state is a vector state where all the entries related to the scheduling vector equal zero, i.e., any stream is scheduled. On the other hand, we consider a state $\mathbf{s}_t$ as terminal if the scheduling vector, $\boldsymbol{\xi}_t$, equals the scheduling vector of the previous state, $\boldsymbol{\xi}_{t-1}$. After reaching a terminal state, the environment resets to an initial state and a new episode starts.

The dimension of the state space vectors is $D_{\text{state}} = KN_{\mathrm{s}} + N_{\mathrm{r}}KN_{\mathrm{t}}$. We assume the state space to be continuous-valued because the entries of $\hat{\mathbf{H}}_{\mathrm{C}}$ can take any complex value, although the entries of $\boldsymbol{\xi}_t$ are binary-valued. Notice that we assume signal processing techniques that handle complex-valued entries. Otherwise, the imaginary and real parts should be treated as independent inputs, leading to vectors twice the size.

The **action** vector $\mathbf{a}_t$ is the binary one-shot encoding representation of the stream to be included in the scheduling of the next state. Hence, the action vector is

$$\mathbf{a}_t = [a_{1,1}, \ldots, a_{K,N_{\mathrm{s}}}]^{\mathrm{T}}$$
$$\text{s.t.} \sum_{k=1}^{K} \sum_{n_{\mathrm{s}}=1}^{N_{\mathrm{s}}} a_{k,n_{\mathrm{s}}} = 1, \tag{12}$$

such that the scheduling vector of the current state, $\boldsymbol{\xi}_t$, and the scheduling vector of the next state, $\boldsymbol{\xi}_{t+1}$, are related as follows: $\boldsymbol{\xi}_{t+1} = \mathrm{OR}(\boldsymbol{\xi}_t, \mathbf{a}_t)$. According to this formulation of actions, the size of the set of feasible actions grows only linearly with the number of streams (i.e., $|\mathcal{A}| = KN_{\mathrm{s}}$). Hence, it overcomes the scalability constraint of previous combinatorial approaches [11], [15], [21]. The dimension of the action vectors is then $D_{\text{action}} = KN_{\mathrm{s}}$.

The **reward** r_t is determined as a function of the sum-rate since it is the metric we aim to maximize. In this formulation, rather than using the sum-rate value itself, we calculate r_t as the difference between the values after and before taking the action $\mathbf{a}_t$, i.e.,

$$r_t = \Lambda_{\boldsymbol{\xi}_{t+1}} - \Lambda_{\boldsymbol{\xi}_t}. \tag{13}$$

The ANNs used for function approximation are sensitive to the scale of the features. If we use the sum-rate values as rewards, the difference between the scales of actions and rewards can be unfavorable for learning and the stability of the consecutive time steps. By computing the rewards as the difference between the sum-rate values, we force the scales of state, action, and reward values to remain similar.

Algorithm 1: PPO Training Algorithm.

1: **Initialize:**
2: set actor ϖ given random $\boldsymbol{\vartheta}_\pi$
3: set critic ν given random $\boldsymbol{\vartheta}_{\mathrm{v}}$
4: **for** $i_{\mathrm{a}} = 0, \ldots, I_{\mathrm{a}} - 1$ **do:**
5: **for each** environment e **do:**
6: create $\mathcal{R}_e$
7: set initial state $\mathbf{s}_0$ with random $\hat{\mathbf{H}}_{\mathrm{C}}$
8: **for** $t = 0, \ldots, T-1$ **do:**
9: get $\mathbf{a}_t \leftarrow \varpi(\mathbf{s}_t, \boldsymbol{\vartheta}_\pi)$
10: get $\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t) \leftarrow \pi(\mathbf{a}_t|\mathbf{s}_t, \boldsymbol{\vartheta}_\pi)$
11: agent performs $\mathbf{a}_t$
12: environment returns r_t, $\mathbf{s}_{t+1}$ and χ_t
13: $\mathcal{R}_e[t] = (\mathbf{s}_t, \mathbf{a}_t, r_t, \chi_t, \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t), \sim, \sim)$
14: **for** $t = T-1, \ldots, 0$ **do:**
15: compute $V_t = (1 - \chi_t)\gamma V_{t+1} + r_t$
16: get $\hat{V}(\mathbf{s}_t) \leftarrow \nu(\mathbf{s}_t, \boldsymbol{\vartheta}_{\mathrm{v}})$
17: compute $A_t = V_t - \hat{V}(\mathbf{s}_t)$
18: $\mathcal{R}_e[t] = (\mathbf{s}_t, \mathbf{a}_t, r_t, \chi_t, \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t), V_t, A_t)$
19: combine all $\mathcal{R}_e$ into $\mathcal{R}$
20: **for** $i_{\mathrm{u}} = 0, \ldots, I_{\mathrm{u}} - 1$ **do:**
21: randomize order of $\mathcal{R}$
22: divide $\mathcal{R}$ into M minibatches $\mathcal{B}_m$
23: **for each** minibatch $\mathcal{B}_m$ **do:**
24: unpack the stored experiences:
25: $\mathcal{E}_i = (\mathbf{s}_i, \mathbf{a}_i, r_i, \chi_i, \pi_{\text{old}}(\mathbf{a}_i|\mathbf{s}_i), V_i, A_i)$
26: **for** $i = 0, \ldots, |\mathcal{B}_m| - 1$ **do:**
27: get $\pi_{\text{upd}}(\mathbf{a}_i|\mathbf{s}_i) \leftarrow \pi(\mathbf{a}_i|\mathbf{s}_i, \boldsymbol{\vartheta}_\pi)$
28: compute $\rho_i = \frac{\pi_{\text{upd}}(\mathbf{a}_i|\mathbf{s}_i)}{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)}$
29: get $\Phi_i \leftarrow$ entropy in ϖ
30: compute L_{CLIP} by using (14)
31: compute H by using (15)
32: compute L_{V} by using (16)
33: compute $L = -L_{\text{CLIP}} - \lambda_{\mathrm{H}} H + \lambda_{\mathrm{V}} L_{\mathrm{V}}$
34: back-propagate the aggregated loss L
35: update $\boldsymbol{\vartheta}_\pi$ and $\boldsymbol{\vartheta}_{\mathrm{v}}$

Output: trained actor (ϖ)

C. The PPO Algorithm

In this section, we present the algorithm used during the offline training of the PPO agent when solving the scheduling optimization problem in (6). In particular, the scheduling policies that maximize the system sum-rate in the considered communication scenarios are learned by following Algorithm 1.

During the initialization stage, the actor and the critic are created with random initial parameters $\boldsymbol{\vartheta}_\pi$ and $\boldsymbol{\vartheta}_{\mathrm{v}}$, respectively. Next, we perform the sample collection stage (lines 4 to 18) where we store the experience tuples that we will use later to train the actor and the critic. We generate the samples by considering the current actor policy running over E parallel environments. The interactions in each environment e are orderly stored in a replay buffer $\mathcal{R}_e$. We consider extended experience tuples whose structure is $(\mathbf{s}_t, \mathbf{a}_t, r_t, \chi_t, \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t), V_t, A_t)$, where $\chi_t \in \{0, 1\}$ equals one if the next state $\mathbf{s}_{t+1}$ is terminal,

and $\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)$ stands for the probability of taking action $\mathbf{a}_t$ in the current state with the current actor policy. When $\chi_t = 1$ (i.e., the next state is terminal), the environment resets and $\mathbf{s}_{t+1}$ becomes an initial state. Let us recall that a new estimated cascaded channel matrix $\hat{\mathbf{H}}_{\text{C}}$ is generated for each initial state, ensuring the exploration of the state space. The state values V_t and the advantage values A_t are computed backward starting from the last visited state as in lines 15 and 17, respectively. Note that for computing these two values, we use the terms in the reduced tuples $(\mathbf{s}_t, \mathbf{a}_t, r_t, \chi_t, \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t), \sim, \sim)$ stored in line 13. We use a conventional and simple approach to the advantage function since no improvement was achieved when evaluating more complex alternatives.

After completing the sample collection stage, we combine all the replay buffers into one ($\mathcal{R}$). Samples in this buffer are used along I_{u} iterations to update the trainable parameters of the actor and the critic networks. At each iteration, the order of the tuples in the buffer $\mathcal{R}$ is randomized as it is no longer relevant. Besides, randomizing reduces the correlation between the samples within a mini-batch, improving the training performance.

In every iteration, we divide the randomized data into M mini-batches ($\mathcal{B}_m, \forall m$). Lines 24 to 35 describe the training process by considering the samples within one mini-batch. For every experience tuple $\mathcal{E}_i$, we determine the probability of taking the action $\mathbf{a}_i$ according to the current actor policy $\pi_{\text{upd}}(\mathbf{a}_i|\mathbf{s}_i)$. Note that this value differs from the stored value $\pi_{\text{old}}(\mathbf{a}_i|\mathbf{s}_i)$ since the policy used for sample collection is transformed along with the update iterations. We compute the ratio between these two probabilities ρ_i as stated in line 28. Next, we compute Φ_i which is the entropy of the probability distribution of the actor for the state $\mathbf{s}_i$.

Finally, by considering all the experience tuples in the mini-batch and the computed values related to them, we determine the individual terms of the joint objective function L. The term L_{CLIP} that we aim to maximize is the main part of the objective function in PPO since it ensures the policy updates remain stable [27], [28]. This term is given by

$$L_{\text{CLIP}} = \frac{1}{|\mathcal{B}_m|} \sum_i \min(\rho_i A_t, \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon) A_t), \quad (14)$$

where ϵ is the hyper-parameter that determines the clipping interval (see [27], [28] for further details on the $\text{clip}(\cdot)$ operator). The other term we aim to maximize is the entropy bonus H which is given by

$$H = \frac{1}{|\mathcal{B}_m|} \sum_i \Phi_i. \quad (15)$$

Including this entropy-related term in the objective function enhances the exploratory behavior of the agent. Finally, we aim to minimize the difference between the estimates computed by the critic and the real state values of the sampled states. Hence, we calculate the critic loss L_{V} as

$$L_{\text{V}} = \frac{1}{|\mathcal{B}_m|} \sum_i (\hat{V}(\mathbf{s}_i) - V_i)^2. \quad (16)$$

Later, we back-propagate the value computed for the joint objective function as $L = -L_{\text{CLIP}} - \lambda_{\text{H}} H + \lambda_{\text{V}} L_{\text{V}}$ and compute the new values of the trainable parameters $\boldsymbol{\vartheta}_\pi$ and $\boldsymbol{\vartheta}_{\text{v}}$ by performing a stochastic gradient descent update. The parameters λ_{H} and λ_{V} stand for adjustable coefficients.

The entire training algorithm runs over I_{a} iterations. The expected result of this algorithm is to obtain a trained actor capable of predicting near-optimal schedulings for unseen channel realizations.

TABLE II
TRAINING CONFIGURATION PARAMETERS

Parameter	Description	Value
I_{a}	algorithm iterations	10000
E	parallel training environments	4
T	sample collection time steps	128
I_{u}	update iterations	4
M	number of mini-batches	8
$\lvert\mathcal{B}_m\rvert$	mini-batch size	64
ϵ	clip coefficient	0.1
λ_{H}	entropy coefficient	0.01
λ_{V}	value function update coefficient	0.5
μ_{c}	optimizer learning rate	0.001
γ	discount factor	0.99

Algorithm 2: PPO Online Scheduling.

Input: trained actor (ϖ)
1: set initial state $\mathbf{s}_0$ with the estimated $\hat{\mathbf{H}}_{\text{C}}$
2: **while** state $\mathbf{s}_t$ **is not TERMINAL** :
3: get $\mathbf{a}_t \leftarrow \varpi(\mathbf{s}_t, \boldsymbol{\vartheta}_\pi)$ (select the stream)
4: take $\mathbf{a}_t$ (include the selected stream)
5: get $\mathbf{s}_{t+1}$ (get the updated scheduling)
6: set $\mathbf{s}_t \leftarrow \mathbf{s}_{t+1}$
Output: scheduling vector $\boldsymbol{\xi}_t$ from terminal $\mathbf{s}_t$

Table II shows the configuration parameters considered for the training of the PPO agent. The selected numbers of algorithm and update iterations provide enough training to reach a good performance of the PPO agent since no significant improvement was observed beyond this point. Using four parallel environments and 128 sample collection steps at each iteration ensures a proper exploration of the state space. Mini-batches with 64 entries constitute an adequate trade-off between complexity and learning speed. We selected the different coefficients, the learning rate, and the discount factor through a grid search approach. The values obtained are similar to those proposed in [27].

The training described in Algorithm 1 is expected to perform mostly offline so that the trained actor can be later deployed in a practical scenario. We next describe in Algorithm 2 the online behavior of the deployed actor. The expected result is to obtain the scheduling vector $\boldsymbol{\xi}_t$ that maximizes the system sum-rate for each estimated cascaded channel $\hat{\mathbf{H}}_{\text{C}}$.

During the online stage, the observed interactions improve scheduling because they allow the trained actor to adapt to changes in the communication environment. However, in this stage, we only compute the system sum-rate for the scheduling vector in the terminal state. Therefore, we must use a different reward assignment, where $r_t = \Lambda_{\boldsymbol{\xi}_t}$ if $\mathbf{s}_{t+1}$ is a terminal state and $r_t = 0$ otherwise. This episodic reward is less efficient than

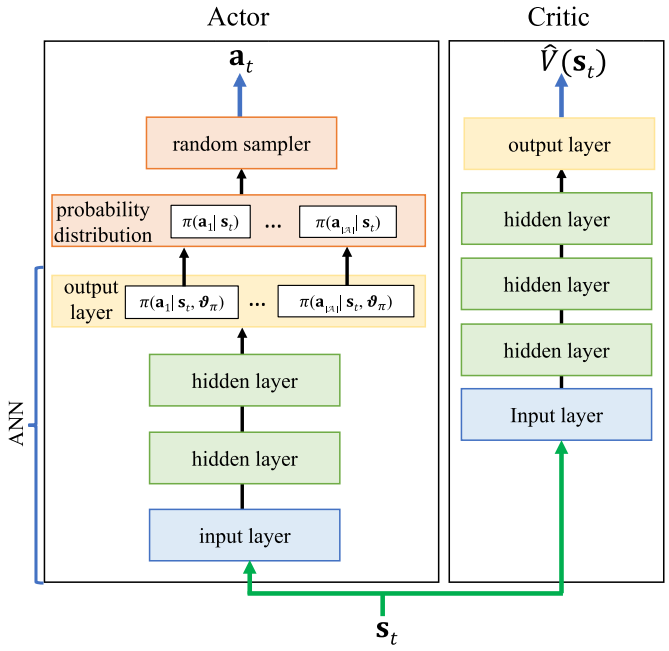


Fig. 4. Actor and critic structures.

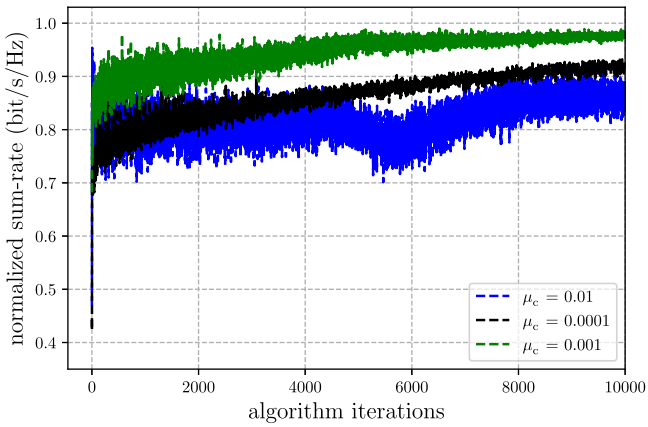


Fig. 5. Normalized sum-rate (bit/s/Hz) vs algorithm iterations.

the one used for offline training but enables the actor to continue learning without sacrificing the scalability and performance of the scheduling algorithm.

Notice that, during online interactions, we select actions deterministically to find the scheduling vectors that maximize system performance. Therefore, we limit the exploration of the action space to avoid unpredictable or undesirable behaviors. As explained in [16], model-based offline training can be deployed simultaneously to ensure sufficient action exploration, such that both kinds of experiences contribute to the learning process. On the other hand, exploration of the state space is guaranteed by the dynamic behavior of the communication conditions.

D. Actor and Critic ANN Structures

Fig. 4 shows the structures of the actor and the critic that we propose to use as parts of the PPO framework. As explained above, the actor comprises three stages: an ANN, a probability distribution stage, and a random sampler. The dimensions of the input and output layers of the actor ANN equal D_{state} and D_{action}, respectively. We use two fully connected hidden layers with $2D_{\text{state}}$ neurons each. We made several tests with different configurations, and no improvement was observed when using bigger setups. We use the rectified linear unit (ReLU) function as activation in the hidden layers. Besides, we use the softmax function at the output layer to obtain the normalized probabilities over the feasible actions.

In the critic ANN, the dimension of the input layer also equals D_{state}. We use three fully connected hidden layers with identical shapes to those in the actor. The output layer dimension is one since this network aims to predict the state-value function for a given state. Hence, we use the linear activation function at this output layer. Finally, we use the Adam optimizer in both actor and critic ANNs since this algorithm has proven to be computationally efficient and robust for supervised and DRL problems [30], [31], [32].

E. Convergence Analysis

As in [8], [33], [34], we perform a convergence analysis to demonstrate the suitability of our proposed algorithm to solve the optimization problem in (6). We begin this evaluation by assessing the system's reward performance. Fig. 5 shows the normalized sum-rate values calculated at the algorithm iterations. Notice that we train a single agent capable of handling various network configurations and channel models. This agent matches the dimensions of the setup employed in most of the simulations considered in Section V. However, we have also considered different setups (such as those with fewer users or varying the number of IRS elements and receiving antennas at the RSU) to assess the generalizability of our proposed solution. To evaluate the convergence metric across different scenarios and channel conditions, we utilize 1000 channel samples that were not visited during the training process. We employ the maximum sum-rate values as normalization factors to enable a general representation encompassing all the simulation setups.

As shown in Fig. 5, the sum-rate values improve and nearly converge for two learning rate configurations. On the considered simulation setups, the best performance is achieved when the learning rate μ_c equals 0.001. These simulation results demonstrate the proper behavior of the actor network since it continuously improves on predicting the scheduling vectors that maximize system performance.

Next, we perform a convergence analysis based on the critic loss (L_V). This parameter measures the difference between the critic network approximation and the actual state-value function, so lower is better. The simulation results in Fig. 6 show the algorithm's convergence with respect to this metric. As in the previous experiment, $\mu_c = 0.001$ offers the best trade-off between convergence speed and stability.

Previous results demonstrate that the PPO-based framework provides reliable and stable solutions for the sequential scheduling problem in the considered scenarios. Both the actor and critic ANNs steadily learn from the interactions and contribute to the system performance in unseen channel realizations. The

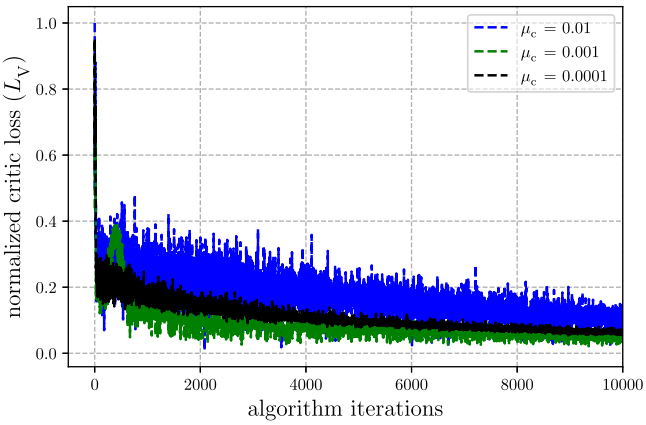


Fig. 6. Normalized critic loss vs algorithm iterations.

convergence of general PPO solutions is also proven in [35], demonstrating the robustness of this approach.

V. Simulation Results

In this section, we present the results of computer simulations, which validate using the PPO framework to find the scheduling vector that maximizes the system sum-rate in the uplink of an IRS-assisted MS MU-MIMO system. We considered the scenarios presented in Sections III-A and III-B, and the following benchmarks:

- User-level exhaustive search (ES): this method evaluates all the 2^K user-level feasible scheduling vectors and exhaustively searches for the one that results in a higher sum-rate value. ES is expensive because the IRS matrix, the precoder matrices, and the resulting sum-rate values must be computed for each possible scheduling vector.
- User-level greedy direct (GD): this method first schedules the vehicle with the highest rate and, in the following steps, schedules the vehicle that enhances the system sum-rate the most. The algorithm stops when no performance improvement can be achieved. GD might require computing the IRS and precoder matrices up to $\frac{K(K+1)}{2} - 1$ times [12, Algorithm 1].
- Stream-level GD: This method first schedules the stream that provides the highest rate and then schedules the stream that enhances the system sum-rate the most in each subsequent step. The algorithm stops when no further performance improvement can be achieved. Stream-level GD may require computing the IRS and precoder matrices up to $\frac{KN_s(KN_s+1)}{2} - 1$ times.
- Greedy indirect (GI): this method starts scheduling the vehicle with the highest cascaded channel norm. In the next steps, it selects the vehicles based on a spatial compatibility metric until reaching $K_{\max}$ vehicles. GI computes the IRS and precoder matrices only for the final scheduling vector [12, Algorithm 2].
- Maximum channel gain (MG): this method schedules sequentially the $K_{\max}$ vehicles with the highest cascaded channel norms. The IRS and precoder matrices are computed only for the final scheduling vector.

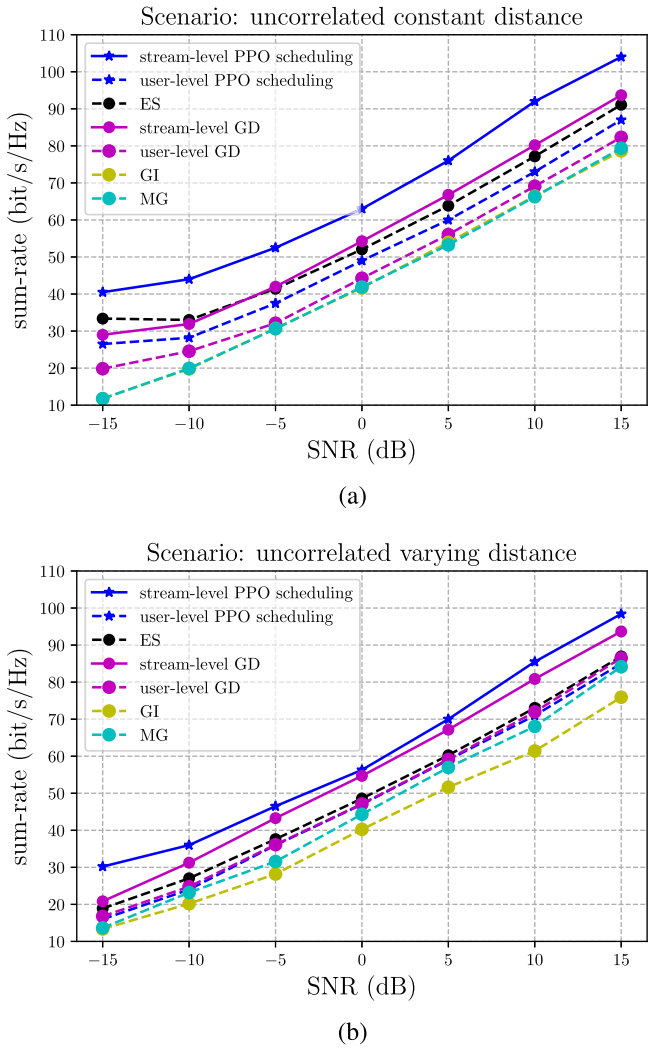
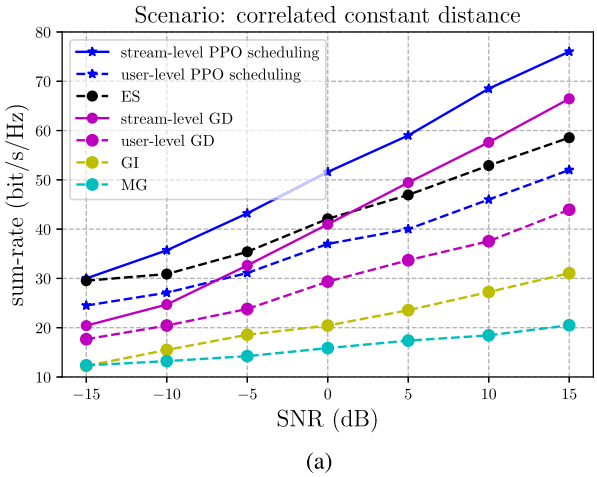


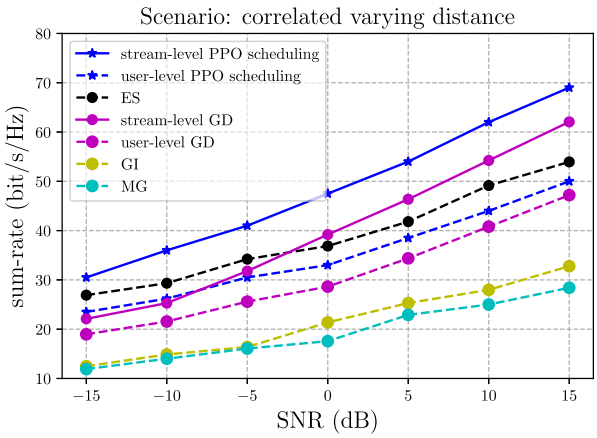
Fig. 7. Sum-rate (bit/s/Hz) vs SNR (dB) in uncorrelated channels for $K = 10,\ N_s = 2,\ N_t = 2,\ N_r = 8,\ N = 30$.

As in [12], we constrain the number of scheduled vehicles in the GI and MG methods to be $K_{\max} = \lfloor \frac{N_r}{N_s} \rfloor$. However, simulation results demonstrate that scheduling more vehicles could lead to higher system sum-rate values in several communication setups. We do not use a stream-level exhaustive search benchmark since the number of feasible scheduling vectors becomes intractable in the considered configurations. For comparison, we include two versions of the proposed PPO-based sequential scheduling, namely the stream-level approach explained in the previous section and a user-level version where if a vehicle is scheduled, all its streams are. To ensure fairness between the algorithms, we use the same approach to compute the IRS and precoder matrices in all cases (i.e., [8, Algorithm 1]).

Figs. 7 and 8 show the achievable sum-rate values obtained in the spatially uncorrelated and correlated channel scenarios, respectively. We considered a setup with $K = 10$ vehicles employing $N_t = 2$ antennas to send $N_s = 2$ streams each, an IRS with $N = 30$ scattering elements, and an RSU with $N_r = 8$ receiving antennas. In the correlated channel scenarios, we set



(a)



(b)

Fig. 8. Sum-rate (bit/s/Hz) vs SNR (dB) in correlated channels for $K = 10,\ N_{\mathrm{s}} = 2,\ N_{\mathrm{t}} = 2,\ N_{\mathrm{r}} = 8,\ N = 30,\ G = 5$.

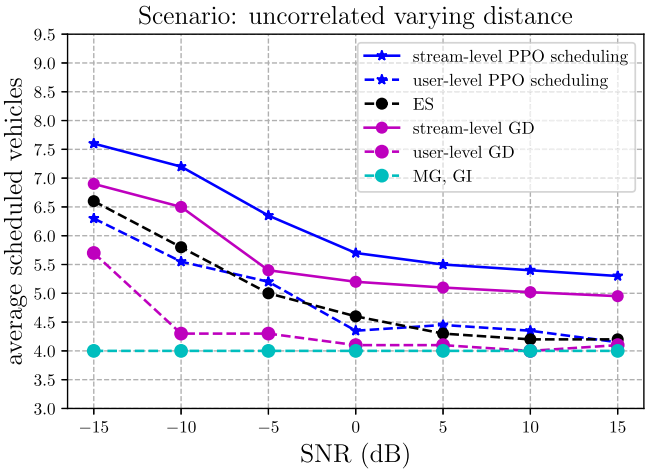


Fig. 9. Average scheduled vehicles vs SNR (dB) in uncorrelated varying distance scenarios for $K = 10,\ N_{\mathrm{s}} = 2,\ N_{\mathrm{t}} = 2,\ N_{\mathrm{r}} = 8,\ N = 30$.

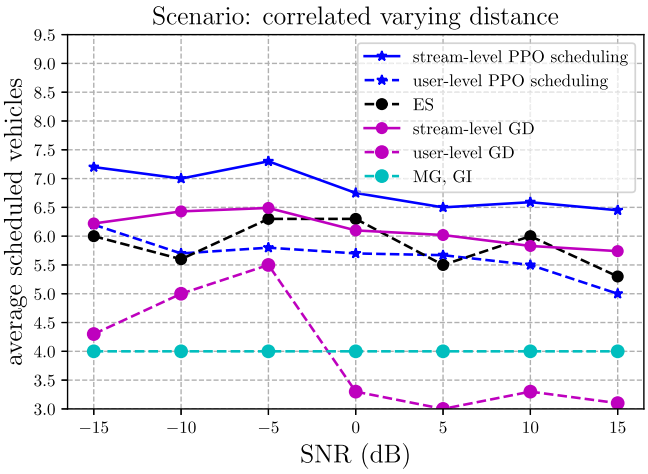


Fig. 10. Average scheduled vehicles vs SNR (dB) in correlated varying distance scenarios for $K = 10,\ N_{\mathrm{s}} = 2,\ N_{\mathrm{t}} = 2,\ N_{\mathrm{r}} = 8,\ N = 30,\ G = 5$.

$G = 5$. The nominal angles for the different groups are selected uniformly in the interval from 0 to $\frac{\pi}{2}$ rad, to avoid mirror angles [22]. As shown, the increment of the sum-rate values is smaller in the correlated scenarios since the interference between the spatially correlated vehicles steeply increases with the SNR values.

In the scenarios considered in Figs. 7 and 8, the stream-level PPO scheduling significantly outperforms the benchmarks and the user-level PPO scheduling. The stream-level PPO scheduling selects the specific streams per vehicle to schedule achieving better interference control. This flexibility is fundamental in scenarios like the ones considered where the number of competing streams is higher than the number of receiving antennas. The stream-level GD benchmark also leverages this capability and outperforms most user-level approaches. Note that GD benchmarks are also sequential. However, they require computing the IRS and precoder matrices for all the evaluated scheduling vectors. These computing costs and delays can limit their usage in rapidly varying vehicular communications.

The gap between stream-level approaches and their user-level counterparts is more evident for high SNR values, specially in the correlated scenarios. In those cases, selecting only specific streams enables to alleviate the inter-vehicle interference and fully leverage the transmission power without affecting other vehicles with similar spatial properties.

The performance of the user-level PPO scheduling is close to that of the ES benchmark and better than more computationally demanding approaches like the user-level GD benchmark. Besides, although the GD approaches are competitive in scenarios where the average channel gains are the key aspect (like in the uncorrelated varying distance scenario), they fail at assessing the long-term effects of the sequential scheduling in more complex setups (like in the correlated scenarios).

The performance of the GI and MG benchmarks is limited by the $K_{\max}$ value, which is lower than the optimal number of scheduled vehicles in several scenarios. Figs. 9 and 10 show the average numbers of vehicles scheduled in the uncorrelated varying distance and correlated varying distance scenarios, respectively. In both setups, the flexibility of the stream-level PPO scheduling enables allocating more vehicles with at least one

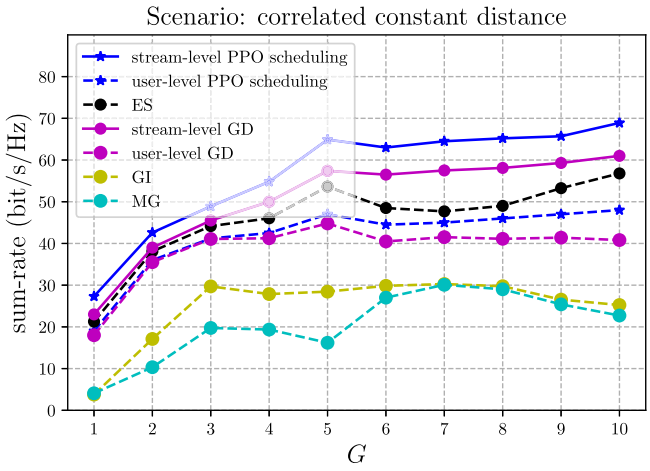


Fig. 11. Sum-rate (bit/s/Hz) vs G in correlated constant distance scenarios for $K = 10,\ N_{\mathrm{s}} = 2,\ N_{\mathrm{t}} = 2,\ N_{\mathrm{r}} = 8,\ N = 30,\ \mathrm{SNR} = 10$ dB.

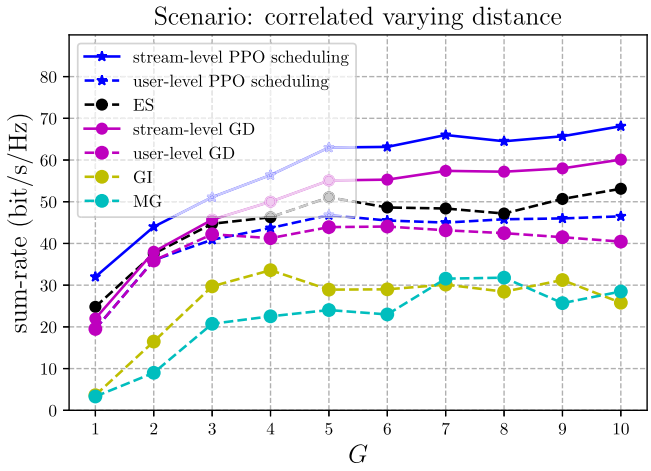


Fig. 12. Sum-rate (bit/s/Hz) vs G in correlated varying distance scenarios for $K = 10,\ N_{\mathrm{s}} = 2,\ N_{\mathrm{t}} = 2,\ N_{\mathrm{r}} = 8,\ N = 30,\ \mathrm{SNR} = 10$ dB.

stream per vehicle without affecting the system sum-rate. Besides, this proposal makes no previous assumption of the number of vehicles to schedule, enabling it to adapt to the characteristics of the several scenarios. User-level PPO schedules an average number of vehicles similar to the ES benchmark. Unlike the GD approaches, both PPO-based algorithms leverage the capability of RL for long-term analysis. The greedy behavior of GD tends to allocate first the vehicles (streams) with the best communication conditions and disregards the effect it has on the final scheduling vector. Because of this, the capability of the user-level GD to schedule more users steeply decreases for large SNR values in both channel configurations. This effect is more evident in the spatially correlated scenarios since vehicles with high channel gains strongly interfere with others of similar spatial properties.

The following experiments enable us to analyze how the spatial correlation affects the scheduling performance. For this purpose, we considered a setup with $K = 10$ vehicles having $N_{\mathrm{t}} = 2$ antennas to send $N_{\mathrm{s}} = 2$ streams each, an IRS with $N = 30$ scattering elements, an RSU with $N_{\mathrm{r}} = 8$ receiving antennas, and an SNR $= 10$ dB. We considered the number of groups of vehicles G ranging from 1 to 10. Note that for $G = 1$, all the vehicles are grouped together and share similar nominal angles and spatial correlation matrices. On the other extreme, for $G = 10$, their spatial properties are well defined.

Figs. 11 and 12 show the results obtained for constant distance and varying distance scenarios, respectively. Again, the GI and MG benchmarks perform poorly since they schedule a predefined number of vehicles ($K_{\max} = 4$) regardless of the communication conditions. For lower values of G, scheduling $K_{\max}$ vehicles leads to significant interference and, therefore, system performance degradation.

As observed in previous experiments, the user-level PPO scheduling and the ES and user-level GD benchmarks perform similarly. The performance of these approaches initially improves with the increase in the number of groups since vehicles can be properly separated. However, from $G = 3$ to $G = 10$, the sum-rate values saturate. Although the nominal angles become more distributed, the degrees of freedom to manage the interference remain limited by the number of receiving antennas at the RSU. The proposed stream-level PPO scheduling reaches the highest performance along the range of G values. Besides, it keeps improving with the increase of G, even in the interval where most benchmarks stagnate.

A. *Imperfect CSI*

In the simulations so far, we have considered perfect CSI (PCSI) to ensure a fair comparison with the different benchmark algorithms since they disregard the effects of channel estimation errors. We next present some computer experiments to illustrate the capability of the proposed stream-level PPO scheduling to address the optimization problem in (6) while considering ICSI.

We evaluated the performance of stream-level PPO scheduling by considering a simulated online stage where channel matrices are estimated with an error of variance σ^2_{com}. We consider this variance σ^2_{com} common to all the estimated combined channel matrices (i.e., $\sigma^2_{\mathrm{com}_n} = \sigma^2_{\mathrm{com}}, \forall n$).

Fig. 13 shows the sum-rate values obtained with the trained PPO-based scheduling agent in a spatially correlated varying distance setup with SNR $= 10$ dB, $K = 10$, $N_{\mathrm{t}} = 2$, $N_{\mathrm{s}} = 2$, $N = 30$, $N_{\mathrm{r}} = 8$, and $G = 5$. The figure also shows the sum-rates of three benchmarks: stream-level PPO with PCSI, user-level ES, and random scheduling.

The results in Fig. 13 show that stream-level PPO outperforms ES in several of the considered scenarios. However, the performance of the proposed solution in ICSI conditions deteriorates for high SNR values because, in this regime, scheduling the wrong vehicles can lead to significant interference. Nevertheless, the proposed solution outperforms the random scheduling benchmark, even for the challenging worst setting ($\sigma^2_{\mathrm{com}_n} = 0.25$ and SNR $= 15$ dB).

During the online stage, the trained agent keeps learning. Although the system performance initially degrades because of the ICSI, it improves by observing up-to-date interactions. This

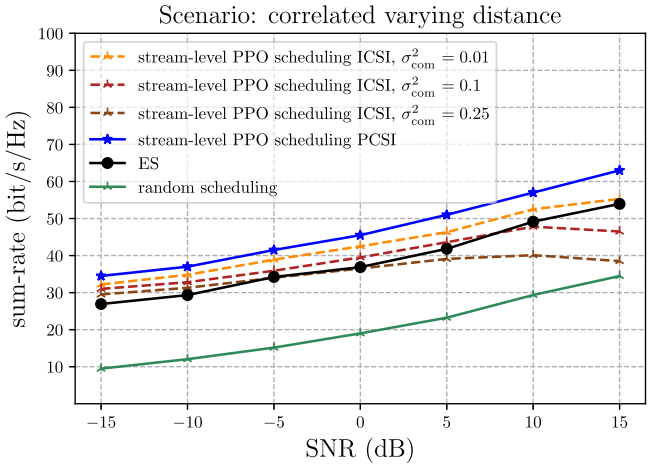


Fig. 13. Sum-rate (bit/s/Hz) vs SNR (dB) in ICSI conditions for SNR = 10 dB, $K = 10$, $N_{\text{t}} = 2$, $N_{\text{s}} = 2$, $N = 30$, $N_{\text{r}} = 8$ and $G = 5$.

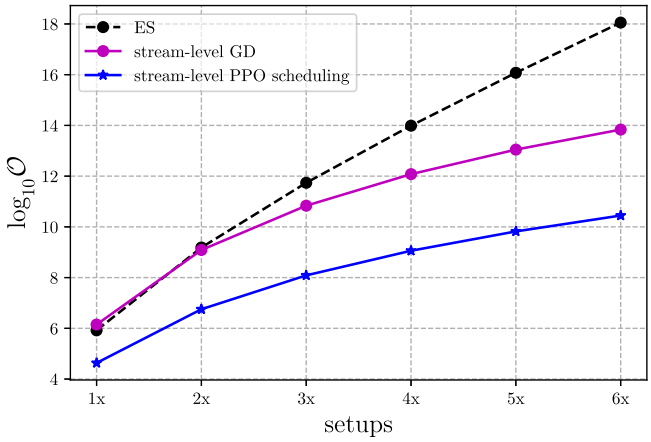


Fig. 14. Computational complexity ($\log_{10}\mathcal{O}$) vs network size.

continuous-learning capability makes our PPO proposal a robust alternative to less flexible classical approaches.

B. Computational Complexity

In this subsection, we present a computational complexity analysis of the stream-level PPO scheduling based on the required multiplications. As stated in [36], this is a high-level metric since it disregards other less time-consuming operations. The complexity analysis for Algorithms 1 and 2, which occur at different moments and have different resource limitations, is addressed separately. Algorithm 1 corresponds to the training stage, which is mostly executed offline, and Algorithm 2 implements the online stage where the trained PPO-based agent is used to predict the best scheduling vectors for the estimated channel matrices.

The highest computational complexity in Algorithm 1 is in the actor and critic ANNs. The computational complexity for ANNs with fully connected layers is bounded by $\mathcal{O}(\varrho\zeta^2)$, where ϱ is the number of hidden layers, and ζ is the number of neurons in the widest layers [34], [36]. We disregard the number of layers ϱ because it does not depend on the communication parameters, and its value is generally small compared to ζ. In both the actor and the critic, ζ equals $2D_{\text{state}}$. Hence, the complexity of the sample collection in Algorithm 1 is in the order of $\mathcal{O}(I_{\text{a}}T(K^2N_{\text{s}}^2 + N_{\text{r}}^2\ K^2N_{\text{t}}^2))$, where I_{a} and T stand for the number of algorithm iterations and sample collection time steps, respectively. On the other hand, the computational complexity during the ANN updates is in the order of $\mathcal{O}(I_{\text{a}}|\mathcal{B}_m|(K^2N_{\text{s}}^2 + N_{\text{r}}^2\ K^2N_{\text{t}}^2))$, where $|\mathcal{B}_m|$ stands for the size of the mini-batches. Finally, the general complexity of Algorithm 1 is in the order of $\mathcal{O}((I_{\text{a}}T + I_{\text{a}}|\mathcal{B}_m|)(K^2N_{\text{s}}^2 + N_{\text{r}}^2\ K^2N_{\text{t}}^2))$.

During the online scheduling stage, the computational complexity is remarkably lower, which is suitable for the stringent latency requirements of vehicular communications. In this stage, the trained actor is used to predict the scheduling vector in a sequential fashion with up to KN_{s} forward passes of the ANN. Since the number of hidden neurons in the actor's widest layers equals $2D_{\text{state}}$, the computational complexity of this stage is in the order of $\mathcal{O}(KN_{\text{s}}(K^2N_{\text{s}}^2 + N_{\text{r}}^2\ K^2N_{\text{t}}^2))$.

Table III compares the computational complexity of the proposed solution to some of the best-performing benchmarks. A significant part of the complexity in these scheduling algorithms is in computing the objective function for a given scheduling vector, i.e., finding the IRS and precoder matrices, and calculating the sum-rate. We reference the complexity of the algorithm used for this purpose in [8, Algorithm 1], which is in the order of $\mathcal{O}(K^2N_{\text{t}}^2\ N^2 + N_{\text{r}}^2\ N^2)$. Notice that our proposed solution performs the IRS and precoder optimization only after finding the right scheduling vector, while both benchmarks evaluate multiple scheduling options.

Fig. 14 shows the number of required multiplications for evaluating several network setups. We start with a simplified configuration (1x) where $K = 5$, $N_{\text{s}} = 2$, $N_{\text{t}} = 2$, $N_{\text{r}} = 4$, and $N = 15$, and gradually increase these values by a linear factor (2x, 3x,..., 6x). As shown in the figure, the computational complexity of the proposed solution is lower in all setups. Moreover, the difference compared to both benchmarks increases with the size of the network parameters. For the largest scenario (6x), the computational complexities of the stream-level GD and ES benchmarks are almost four and eight orders of magnitude larger, respectively.

VI. Conclusion

We have investigated a sequential DRL-based scheduling approach for the sum-rate maximization in the uplink of an IRS-assisted MU MIMO communication system. The optimization problem is formulated as a POMDP, and a PPO-based framework is proposed to address the scheduling task in a sequential fashion. The scheduling capabilities have been extended to the stream level, and this proposal has been tested in several scenarios with spatially correlated and uncorrelated channels and ICSI conditions. We assessed the proposed stream-level PPO-based sequential scheduler against several benchmarks. Our main findings can be summarized as follows:

- The sequential DRL-based scheduling formulation enables us to overcome the scalability limitations of combinatorial

TABLE III
COMPUTATIONAL COMPLEXITY

Algorithm	Computational Complexity
Stream-level PPO scheduling	$\mathcal{O}(KN_{\mathrm{s}}(K^2N_{\mathrm{s}}^2+N_{\mathrm{r}}^2K^2N_{\mathrm{t}}^2)+(K^2N_{\mathrm{t}}^2N^2+N_{\mathrm{r}}^2N^2))$
Stream-level GD	$\mathcal{O}((\frac{KN_{\mathrm{s}}(KN_{\mathrm{s}}+1)}{2}-1)(K^2N_{\mathrm{t}}^2N^2+N_{\mathrm{r}}^2N^2))$
ES	$\mathcal{O}(2^K(K^2N_{\mathrm{t}}^2N^2+N_{\mathrm{r}}^2N^2))$

approaches, as it reduces the number of feasible actions from $2^{KN_{\mathrm{s}}}$ to KN_{s}.

- The proposed scheduler outperforms the considered benchmarks in terms of system sum-rate in all the evaluated scenarios. Besides, it allows scheduling more vehicles by selecting the appropriate streams to allocate.
- The proposed solution has proven robust in ICSI conditions since it achieves a competitive performance regarding the optimal user-level scheduling even for high estimation errors

REFERENCES

[1] S. Gyawali, S. Xu, Y. Qian, and R. Q. Hu, "Challenges and solutions for cellular based V2X communications," *IEEE Commun. Surveys Tut.*, vol. 23, no. 1, pp. 222–255, Firstquarter 2021.

[2] Z. Zhang et al., "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, Sep. 2019.

[3] V. Va, T. Shimizu, G. Bansal, and R. W. Heath Jr., "Millimeter wave vehicular communications: A survey," *Foundations Trends Netw.*, vol. 10, pp. 1–113, 2016.

[4] Y. Zhu, B. Mao, and N. Kato, "Intelligent reflecting surface in 6G vehicular communications: A survey," *IEEE Open J. Veh. Technol.*, vol. 3, pp. 266–277, 2022.

[5] M. Alsabah et al., "6G wireless communications networks: A comprehensive survey," *IEEE Access*, vol. 9, pp. 148191–148243, 2021.

[6] Y. Cao, S. Xu, J. Liu, and N. Kato, "Toward smart and secure V2X communication in 5G and beyond: A UAV-Enabled aerial intelligent reflecting surface solution," *IEEE Veh. Technol. Mag.*, vol. 17, no. 1, pp. 66–73, Mar. 2022.

[7] G. Geraci et al., "What will the future of UAV cellular communications be? A flight from 5G to 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1304–1335, Thirdquarter 2022.

[8] D. Pereira-Ruisanchez, O. Fresnedo, D. Perez-Adan, and L. Castedo, "Deep contextual bandit and reinforcement learning for IRS-Assisted MU-MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9099–9114, Jul. 2023.

[9] M. Noor-A-Rahim, Z. Liu, H. Lee, G. G. M. N. Ali, D. Pesch, and P. Xiao, "A survey on resource allocation in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 701–721, Feb. 2022.

[10] A. Alwarafy, M. Abdallah, B. S. Çiftler, A. Al-Fuqaha, and M. Hamdi, "The frontiers of deep reinforcement learning for resource management in future wireless HetNets: Techniques, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 322–365, 2022.

[11] D. Sandberg, T. Kvernvik, and F. D. Calabrese, "Learning robust scheduling with search and attention," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1549–1555.

[12] E. Castañeda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user MIMO systems," *IEEE Commun. Surveys Tut.*, vol. 19, no. 1, pp. 239–284, Firstquarter 2017.

[13] I. A. Bartsiokas, P. K. Gkonis, D. I. Kaklamani, and I. S. Venieris, "ML-Based radio resource management in 5G and beyond networks: A survey," *IEEE Access*, vol. 10, pp. 83507–83528, 2022.

[14] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges," *IEEE Commun. Surveys Tut.*, vol. 22, no. 2, pp. 1251–1275, Secondquarter 2020.

[15] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning Series)*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[17] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds. 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[18] R. Huang and V. W. Wong, "Neural combinatorial optimization for throughput maximization in IRS-Aided systems," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.

[19] R. Huang and V. Wong, "Joint user scheduling, phase shift control, and beamforming optimization in intelligent reflecting surface-aided systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7521–7535, Sep. 2022.

[20] M. Joham, H. Gao, and W. Utschick, "Estimation of channels in systems with intelligent reflecting surfaces," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5368–5372.

[21] A. Al-Hilo, M. Samir, M. Elhattab, C. Assi, and S. Sharafeddine, "Reconfigurable intelligent surface enabled vehicular communication: Joint user scheduling and passive beamforming," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2333–2345, Mar. 2022.

[22] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *FNT Signal Process.*, vol. 11, no. 3/4, pp. 154–655, 2017. [Online]. Available: http://www.nowpublishers.com/article/Details/SIG-093

[23] R. Hunger, M. Joham, and W. Utschick, "On the MSE-Duality of the broadcast channel and the multiple access channel," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 698–713, Feb. 2009.

[24] Q. Wu and R. Zhang, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1838–1851, Mar. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8930608/

[25] K. Stylianopoulos, G. Alexandropoulos, C. Huang, C. Yuen, M. Bennis, and M. Debbah, "Deep contextual bandits for orchestrating multi-user MISO systems with multiple RISs," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1556–1561.

[26] J. Zhang, L. Dai, X. Zhang, E. Björnson, and Z. Wang, "Achievable rate of Rician large-scale MIMO channels with transceiver hardware impairments," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8800–8806, Oct. 2016.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Aug. 2017, *arXiv: 1707.06347*.

[28] D. Bick, "Towards delivering a coherent self-contained explanation of proximal policy optimization," Master's thesis, University of Groningen, The Netherlands, 2021. [Online]. Available: https://fse.studenttheses.ub.rug.nl/25709/

[29] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1008–1014. [Online]. Available: https://papers.nips.cc/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[30] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1839–1850, Aug. 2020.

[31] K. Feng, Q. Wang, X. Li, and C.-K. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 745–749, May 2020.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds. 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[33] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan. 2021.

[34] Y. Zhao, I. G. Niemegeers, and S. M. H. D. Groot, "Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods," *IEEE Access*, vol. 9, pp. 102953–102965, 2021.
[35] M. Holzleitner, L. Gruber, J. Arjona-Medina, J. Brandstetter, and S. Hochreiter, "Convergence proof for actor-critic methods applied to PPO and RUDDER," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII: Special Issue In Memory of Univ. Prof. Dr. Roland Wagner (Lecture Notes in Computer Science Series)*, A. Hameurlain and A. M. Tjoa, Eds., Berlin, Germany: Springer, 2021, pp. 105–130, doi: 10.1007/978-3-662-63519-3_5.
[36] P. J. Freire, S. Srivallapanondh, A. Napoli, J. E. Prilepsky, and S. K. Turitsyn, "Computational complexity evaluation of neural network applications in signal processing," 2022. [Online]. Available: http://arxiv.org/abs/2206.12191

Dariel Pereira-Ruisánchez (Student Member, IEEE) received the B.S. degree in telecommunications and electronics engineering from the Technological University of Havana José Antonio Echeverría (CUJAE), Havana, Cuba, in 2017. He is currently working toward the Ph.D. degree with the University of A Coruña (UDC), Spain. Since 2021, he has been with the CITIC, Centre for Information and Communications Technology Research and the Group of Electronic Technology and Communications, UDC. His research interests include wireless communication systems, deep learning-based signal processing techniques, and broadcast applications. He was the recipient of the Marie Skłodowska-Curie Predoctoral Fellowship within the 3-i ICT Ph.D. Programme hosted by CITIC.

Óscar Fresnedo (Member, IEEE) received the Computer Engineering and the Ph.D. degree in computer engineering from the University of A Coruña, Spain, in 2007 and 2014, respectively. Since 2007, he has been with the Group of Electronic Technology and Communications (GTEC), Department of Electronics and Systems, University of A Coruña, where he had the benefit of an FPI scholarship granted by the Spanish Government from 2008 to 2012. He has authored or coauthored 20 papers in international technical journals as well as more than 30 papers in relevant international conferences and workshops in the area of communications and signal processing. His main research interests include coding schemes, analog joint source-channel coding, multi-user communications, and image processing. He has participated as a research member in more than 20 research projects and contracts granted by regional, national and European administrations. Dr. Oscar Fresnedo was the recipient of the Best Student Paper Award at the 14th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Darmstadt, 2013.

Darian Pérez-Adán (Member, IEEE) received the B.S. degree in telecommunications and electronics engineering from the Technological University of Havana José Antonio Echeverría, Cuba, in 2017, and the Ph.D. degree in computer engineering from the University of A Coruña (UDC), Spain, in 2022. Since 2018, he has been with the Group of Electronic Technology and Communications, UDC. He was a Visiting Researcher with the Associate Institute for Signal Processing, Technische Universität München (TUM), Munich, Germany, in 2021. His research interests include signal processing for millimeter-wave and multiuser communications. He was the recipient of the a Predoctoral scholarship granted by the Spanish Government.

Luis Castedo (Senior Member, IEEE) received the Ph.D. Telecommunications Engineering degree from the Technical University of Madrid, Madrid, Spain, in 1993. Since 1994, he has been a Faculty Member with the Department of Computer Engineering, University of A Coruña (UDC), Spain, where he became a Professor in 2001 and acted as the Chairman between 2003 and 2009, and again from 2021 to the present. He had previously held several research appointments with the University of Southern California, Los Angeles, CA, USA, and École supérieure d'électricité, Gif-sur-Yvette, France. He has coauthored more than 300 papers in peer-reviewed international journals and conferences. His research interests include signal processing, communication theory and prototyping of terminal equipment for wireless communications. He has also been the Principal Researcher of more than 50 research projects funded by public organisms and private companies. Between 2014 and 2018, he has been Manager of the Communications and Electronic Technologies (TEC) program in the State Research Agency of Spain. He has been General Co-Chair of the 8th IEEE Sensor Array and Multichannel Signal Processing Workshop in 2014 and the 27th European Signal Processing Conference in 2019. He was the recipient of three best paper awards at international conferences and the research medal Isidro Parga Pondal awarded by the Royal Galician Academy of Sciences in 2021.