On Coordinated Scheduling of Radio and Computing Resources in Cloud-RAN

Mahdi Sharara, Sahar Hoteit, Patrick Brown, Véronique Vèque

On Coordinated Scheduling of Radio and Computing Resources in Cloud-RAN

Mahdi Sharara, *Student Member, IEEE*, Sahar Hoteit, *Member, IEEE*, Patrick Brown, *Member, IEEE* and Véronique Vèque, *Member, IEEE*

***Abstract*—Cloud Radio Access Network is a promising mobile network architecture based on centralizing the baseband processing of many cellular base stations in a BBU (BaseBand Unit) pool. Such architecture has many advantages. However, computing resources are shared among the base stations connected to the BBU pool. It is challenging to schedule the processing of users' data, especially on overloaded BBU pools, while respecting the time constraints imposed by the Hybrid Automatic Repeat Request (HARQ) mechanism. Given that the processing time of users' data and the computing requirement depends on the radio parameters such as the Modulation and Coding Scheme (MCS), we propose to enable the coordination between radio and computing resources schedulers; such coordination makes the selection of MCS dependent on the availability of radio and computing resources and on the ability to process data while respecting the HARQ-deadline. In this context, we propose and evaluate three Integer Linear Programming (ILP)-based schemes and three low-complexity heuristics, demonstrating their ability to reduce the wasted transmission power. Moreover, we evaluate the performance of the coordination under a multi-services scenario consisting of two services having heterogeneous requirements, enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low-Latency Communication (URLLC).**

***Index Terms*—Cloud-RAN, 5G, Computing Resources Allocation, Integer Linear Programming (ILP), Modulation and Coding Scheme (MCS)**

I. Introduction

Cloud Radio Access Network (C-RAN) is a key pillar in future Mobile Networks. It consists in decoupling Base Band Units (BBUs) from Radio Remote Heads (RRHs) and centralizing the baseband processing of many Radio Remote Heads (RRHs) in a shared BBU pool [2]. The latter processes some virtualized functions such as fast Fourier transform, demodulation, and decoding, among others. The architecture of Cloud-RAN is shown in Fig. 1.

On the one hand, decoupling baseband processing from radio elements in C-RAN leads to multiple advantages. It reduces CAPEX and OPEX of network operators, eases the implementation of interference management mechanisms, increases flexibility and energy efficiency, and improves user experience [4]. On the other hand, computing resources of the BBU pool may become limited as they are shared among a large number of Radio Remote Heads (RRHs) connected to the BBU pool. It is necessary to efficiently manage the BBU pool, especially when it is overloaded with many RRHs. In such a case, the BBU pool has to ensure it maintains the ability to process users' data before exceeding the deadline imposed by the MAC-layer mechanism, known as Hybrid Automatic Repeat Request (HARQ). Knowing that the processing times of users' data depend on radio parameters such as the Signal to Interference plus Noise Ratio (SINR) and the Modulation Coding Scheme index (MCS), the choice of the MCS index affects the users' throughput by specifying the modulation and the code rate to be used [5] [6]. This raises the coordination between radio and computing schedulers as a candidate to efficiently manage the resources of an overloaded BBU pool. This coordination should help achieve the required quality of service (QoS) for users and ensure a good level of fairness.

Throughout this work, we propose different schemes that implement coordination between radio and computing schedulers in Cloud-RAN. These coordination schemes permit the adjustment of users' MCS indexes to ensure the processing of their data in the BBU pool. The processing time of data increases as the MCS index increases [6]. Thus, when the BBU pool gets overloaded, it will not be able to process all users' data while respecting the HARQ deadlines requirements

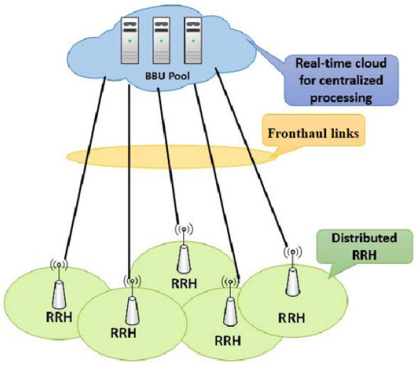


Fig. 1: Cloud-RAN architecture [3]

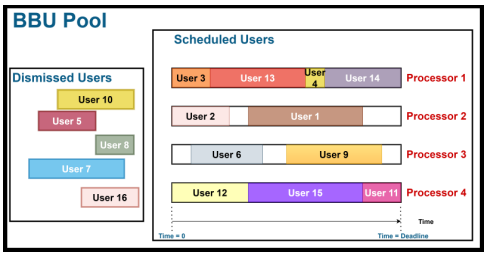


Fig. 2: Scheduling the processing of users in the BBU pool. For an overloaded BBU pool, some users will be dismissed

This research work has been partially carried out in the framework of IRT SystemX, Paris-Saclay, France, and has been granted with public funds within the scope of the French Program "Investissements d'Avenir". Also, it has been partially carried out in the framework of ANR HEIDIS (https://heidis.roc.cnam.fr/; ANR-21-CE25-0019) project.

This work was partially presented at 2021 IEEE/IFIP Integrated Management (IM) conference [1].

The authors M. Sharara, S. Hoteit, and V. Vèque are with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France. (emails: {firstname.lastname}@universite-paris-saclay.fr).

Patrick Brown's email: brown.patrick2@gmail.com

(Fig. 2). Users of non-processed data should retransmit the data as part of the HARQ mechanism. Such retransmission turns out to be an energy-inefficient phenomenon that reduces network performance and wastes radio resources. Hence, instead of retransmitting the data, it could be more efficient to reduce data processing times by decreasing the corresponding MCS indexes of users. While lowering the MCS index will decrease the radio throughput and the required processing time, it should guarantee the ability to process users' data instead of dropping them.

In this context, we aim to study and evaluate some coordination schemes that allow the radio scheduler to assign users' MCS indexes based not only on the radio quality but also on the availability of computing resources. The MCS assignment should consider the availability of computing resources in the BBU pool, and whether they are sufficient to process users' data given the deadline constraints. We note that the radio scheduler can only adjust the users' MCS indexes to values lower than the maximum allowed indexes; the latter is computed according to the radio conditions. This means a user cannot use an MCS index higher than the maximum allowed one; otherwise, the probability of erroneous transmissions increases, leading to decoding failure at the receiver.

In this paper, we investigate three Integer Linear Programming (ILP) coordination solutions, namely, Maximize Total Throughput (MTT), Admit All Users (AAU), and Maximize Users' Satisfaction (MUS), where the selection of users to be scheduled, along with their corresponding MCS indexes, is based on the adopted strategy. We compare and evaluate their performance according to different metrics such as the total throughput, the number of admitted users, fairness, and wasted power. Knowing that the ILP problems are NP-Hard [7], we propose three low-complexity heuristics that depict the performance of the ILP solutions. The results provide insights to network operators on what policy to select depending on the metric they prioritize.

Furthermore, the fifth-generation (5G) of mobile communications has been designed to accommodate different coexisting services with heterogeneous requirements such as enhanced Mobile Broadband (eMBB) and Ultra-Reliable Low-Latency Communication (URLLC) scenarios [8]; so we consider multi-services. While eMBB aims at achieving very high data rates, URLLC supports low-latency transmissions with very high reliability. Once URLLC frames arrive at the BBU pool, they should be scheduled without delay due to the strict latency requirement. Hence the BBU pool may preempt the processing of eMBB data and process URLLC data instead. To implement the proposed coordination, a mobile operator would go for the low-complexity heuristics. For that, we analyze the performance of the heuristics in a multi-services scenario when the existence of URLLC traffic harms eMBB users.

The main contributions of this paper are summarized here:

1) We propose a coordination scheme between radio and computing schedulers that considers the availability of computing resources when assigning MCS indexes for transmissions.
2) We propose three ILP-based algorithms that enable coordination: Maximize Total Throughput (MTT), Admit All Users (AAU), and Maximize Users' Satisfaction (MUS).
3) We analyze the performance of the coordination algorithms with respect to no-coordination counterparts.
4) We propose heuristics that perform close to the ILP-based algorithms but significantly reduce execution time.
5) We analyze the performance of the coordination policies when eMBB and URLLC services coexist, and we study how prioritizing URLLC frames over eMBB ones affects the performance of the heuristics.

The rest of this paper is organized as follows: Section II presents the state of the art. The proposed coordination solutions are presented in section III and evaluated in IV. The proposed heuristics are evaluated in section V. The performance of coordination in a multi-service environment is evaluated in VI. Finally, the work is concluded in section VII.

II. Related Work

Cloud-RAN continues to receive much interest in many research works [4]–[6], [9]–[13]. Some papers have considered the different variations of Cloud-RAN architecture and considered resource allocation problems for these architectures, including the standard Cloud-RAN, Heterogeneous Cloud-RAN with High Power Nodes and Low Power Nodes, Virtualized Cloud-RAN, and Fog RAN [4]. The different resource allocation problems aim at optimizing the resource allocation considering one or more objectives; minimizing energy consumption, minimizing CAPEX/OPEX, maximizing throughput, and minimizing latency as [4] shows.

In [5], the authors study the different processing times of BBU uplink functions and show that the decoding function is the largest consumer of computational resources. The authors show that Fast Fourier Transform (FFT) and demodulation functions do not depend on the MCS index and require less processing time compared to the decoding function, whose processing time increases with the MCS index. Besides, the authors develop some models for processing time prediction using interpolation and deep learning techniques. A better processing time model has been proposed in [9]. Unlike the model in [5], which provides the processing time as a function of the MCS index, the work in [9] models the processing time as a function of the MCS index, the number of RBs, and the CPU frequency. The authors in [10] study the effect of applying the decoding function in parallel and propose two algorithms that ensure parallelism. Their results show that such an option has a good impact by reducing the run time of the decoding function. In [6], the authors propose two different computing scheduling algorithms to schedule the processing of RRHs' data arriving at every transmission time interval (TTI). These algorithms aim to increase the number of correctly decoded sub-frames and the system throughput. The authors evaluate the performance of these algorithms as a function of the number of RRHs assigned to the BBU pool. Additionally, they compare their proposed algorithms to known scheduling heuristics that only focus on computing scheduling without considering radio scheduling. These papers, in contrast with ours, consider neither the multi-services scenarios nor the coordination between the radio and the computing schedulers as our paper does.

The authors in [11] and [12] model the problem of RRH-BBU association as a potential game and a coalition game, respectively. Both papers formulate the BBUs' power consumption in terms of the radio throughput and aim at minimizing it. In the same context, the authors of [14] considered radio resource allocation followed by RRH-BBU association aiming at minimizing power consumption. Their problem aims to minimize the number of BBUs used to process computational requirements. They also proposed two heuristic algorithms to solve these two problems. In [15], the user-RRH association problem has been considered aiming at minimizing interference without considering the computing resources allocation. In contrast, the authors of [16] considered the problem of UE-RRH association followed by the BBU processing allocation.

They formulated a Bin-Packing algorithm to decide which BBUs should process users' data, and then they proposed lower-complexity algorithms to solve these two problems. In [13], the authors consider the issue of joint radio and computing resources allocation in Cloud-RAN. They develop a two independent steps approach that firstly works on allocating computing resources by mapping users to BBUs running as virtual machines; each virtual machine is a BBU. After mapping users to BBUs, the radio resource allocation is carried out by controlling each user's beamforming vector and the power. The authors use less accurate models, in comparison with our practical models, to calculate the achievable throughput and the required computing resources. Instead of showing how much joint allocation is beneficial in comparison with non-joint, they only model the problem and devise two algorithms to allocate the radio and computing resources. In the same context, the authors in [17] investigate the communication and computing resource allocation and formulate the problem using queuing theory. They propose an optimization problem that tries to ensure the stability of RRHs and BBU queues by controlling the assignment of users to RRHs and the assignment of RRHs to BBUs in a way that decreases the response time. The problem is solved by an auction-based algorithm. While this paper proposes a sequential allocation for radio and computing resources allocation problem, our work models the coordination problem, which permits feedback between radio and computing schedulers. Besides, our paper compares the case where the radio and computing scheduler coordination is enabled to the case where it is not.

The performance of coexisting eMBB and URLLC services has been considered in several papers [18]–[20]. In [18], a bankruptcy game has been proposed to allocate radio resources to eMBB and URLLC services. The authors in [19] study the user-plane latency of URLLC for different numerologies and consider the impact of allocating fixed bandwidth for URLLC service on URLLC users' latency and on eMBB users' throughput. Moreover, a Deep Reinforcement Learning (DRL) approach was used in [20] to learn a policy that preserves the quality of service required for eMBB and URLLC users. The reward function was designed to encourage actions that decrease the latency for URLLC users and increase throughput for eMBB users. Inspired by these papers, we consider the hybrid scenario where eMBB and URLLC services coexist and may affect the performance of each other negatively. This would allow us to understand if the existence of URLLC users may worsen the performance of the coordination or not.

To the best of our knowledge, we are the first to consider a coordination scheme between radio and computing schedulers in Cloud-RAN that consists in adjusting users' MCS index and to study its performance in a hybrid eMBB/URLLC environment.

This work is an extension of our previously published work [1]. The paper in [1] is restricted to studying the performance of the proposed coordination when only eMBB users are considered. Additionally, it uses a less accurate processing time model. Moreover, it lacks a complexity analysis of the proposed coordination heuristics, and it does not show how much execution time can be reduced with respect to the ILP problems. In contrast, our current study uses a more accurate processing time model, evaluates the reduction in execution time of the heuristic compared to their ILP counterparts, and studies the performance of the coordination in a multi-services environment consisting of URLLC and eMBB services.

III. Context and problem formulation

The system under study consists of a set of RRHs connected to a centralized BBU pool composed of homogeneous CPU cores with the same execution speed. We suppose each RRH has one antenna. Modifying the number of antennas should not affect the tendency of our results, except that it would only overload the BBU pool at a lower number of RRHs. Furthermore, we assume there is no bottleneck at the fronthaul links connecting the RRHs to the BBU pool. As the uplink processing time is at least 7 times larger than that in downlink [5], it is thus a dominating issue for the BBU pool's bottleneck. For that, we focus on the uplink direction where users connected to each RRH share the available resource blocks that can be used for transmission at the start of every transmission time interval TTI. The RRHs send the received users' data to the BBU pool, which has to process all the incoming data from the RRHs' users within a specified amount of time equal to $2ms$, as instructed by (HARQ)[1] mechanism, and the acknowledgment should be delivered to users in $8ms$ [10].

We further consider that a user's MCS index is determined by jointly considering the channel conditions of all the RBs in the associated RRH. This allows the radio scheduler to attribute the same modulation and coding scheme (MCS) index to a given user over all its resource blocks. To fully focus on the benefits of this proposal, we suppose that radio-related decisions (i.e., including RB allocation, power, MCS indexes, interference management, and frequency spectrum reuse) will have been implicitly managed by the radio scheduler in advance. Hence our model does not intervene in radio scheduling decisions except for allowing the computing scheduler to send a feedback to the radio scheduler regarding the modification of the MCS indexes. It is worth mentioning that the radio scheduler attributes the maximum allowed MCS index i to a given user by considering its radio conditions measured by the user's equipment. More specifically, the Channel Quality Indicator (CQI), which is related to the Signal-to-Noise-and-Interference ratio, is sent by the user equipment (UE); the CQI carries information on how good/bad the communication channel quality is [5]. Based on this indicator, the radio scheduler determines the maximum allowed Modulation Coding Scheme (MCS) index for each user. As shown in [5], the processing time of the BBU sub-functions (more particularly, the decoding function) strongly depends on the MCS index; it increases with the increase of the MCS index. Hence, if the BBU pool is overloaded and all users use their maximum allowed MCS index, the BBU pool will fail to process all the incoming users' data by the specified deadline. We note that if the BBU pool fails to deliver the HARQ-acknowledgment before the deadline, users of non-processed data should retransmit the data.

Next, we present three Integer-Linear-Programming coordination solutions, each with a different objective to maximize.

A. Notations

Let $\mathcal{R}$ be the set of RRHs, $\mathcal{U}_r$ the set of users connected to RRH r, $\mathcal{M}$ the set of possible MCS indexes that can be assigned for the radio transmission by the radio scheduler, and $\mathcal{C}$ be the set of homogeneous CPU cores in the BBU pool. For each RRH r, the coordination policy must attribute to each

[1]In HARQ, the data sent from a user need to be transmitted, received, processed, and acknowledged by the BBU, and the sender should receive the acknowledgment in no more than $8ms$. Hence, the deadline for completing the BBU processing of user's data in the uplink is equal to 2ms after deducting the expected latency in fronthaul, transmission, acquirement, etc.

TABLE I: Summary of the general notations

Parameters	***Definition***
$\mathcal{R}$	Set of RRHs
$\mathcal{U}_r$	Set of users for each RRH $r \in R$
$\mathcal{M}$	Set of MCS indexes that can be used in the system
$\mathcal{C}$	Set of CPU cores in the shared BBU pool (multi-core data center).
$M_{r,u,max}$	Maximum MCS index user $u \in \mathcal{U}_r$ may use
$t_{r,u,m}$	Data processing time of user $u \in \mathcal{U}_r$ having an MCS index $m \in \mathcal{M}$
$b_{r,u,m}$	Data length (in bits) of user $u \in \mathcal{U}_r$ using an MCS index $m \in \mathcal{M}$ during one TTI
$b_{r,u,max}$	Data length (in bits) of user $u \in \mathcal{U}_r$ using its maximum MCS index $M_{r,u,max}$ during one TTI
d	Processing time deadline
$x^c_{r,u,m}$	A binary variable that assigns the data of user $u \in \mathcal{U}_r$ having an MCS index m to the core $c \in \mathcal{C}$
P_t	The maximum transmission power for each user.
$SysMCS$	Adjustable system-wide MCS index, no user is allowed to use a higher one
$MaxMCS$	$\max(\{M_{r,u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$
$selectedMCS_{r,u}$	Selected MCS index for user $u \in \mathcal{U}_r$
$selectedCPU_{r,u}$	Selected CPU to process the data of user $u \in \mathcal{U}_r$
$AdjMargin$	Adjustment Margin; sets a limit on how much the MCS index can be adjusted
$AvTime(c)$	Available processing time on CPU $c \in \mathcal{C}$
Q_m	The number of bits per symbol.
$N^{RE}_{MiniSlot}$	the number of Resource Elements (OFDM symbols) per a mini-slot per 1 sub-carrier
N^{SC}_{RB}	the number of sub-carriers per a Resource Block
OH	Transmission Overhead of control data

user $u \in \mathcal{U}_r$ an MCS index $m \in \mathcal{M}$ that is lower or equal to the maximum allowed MCS index which would initially be chosen by the radio scheduler $M_{r,u,max}$. Based on the selected index m, user u transmits an amount of data that is equal to $b_{r,u,m}$; the latter is determined according to [21] that maps the transport block size (i.e., the payload that can be carried by the physical layer) to the modulation coding scheme index and the number of resource blocks. Besides, the time required for processing user's $u \in \mathcal{U}_r$ data on the BBU pool is equal to $t_{r,u,m}$; the latter is determined using the formula in [9]. We suppose that each user transmits its data using its maximum allowed transmission power P_t [22]. Overall, this refers to the worst-case scenario regarding total radio power consumption. Table I presents the notations used throughout the paper.

B. The coordination solutions for radio and computing scheduling

To model the proposed coordination solutions, we consider three Integer Linear Programming (ILP) optimization problems in which the coordination management entity acts as a single centralized decision-maker.

1) *Maximize Total Throughput (MTT)*: As one of the major objectives in 5G networks is to provide high overall throughput, the first solution we examine tackles this issue by solving the following ILP optimization problem:

$$\text{maximize} \quad \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}}\sum_{c\in\mathcal{C}} x^c_{r,u,m} b_{r,u,m} \tag{1}$$

$$\text{subject to} \quad x^c_{r,u,m} \in \{0,1\},\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C} \tag{2}$$

$$\sum_{c\in\mathcal{C}}\sum_{m\in\mathcal{M}} x^c_{r,u,m} \le 1,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{3}$$

$$x^c_{r,u,m} = 0,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, c \in \mathcal{C}, m > M_{r,u,max}, \tag{4}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}} x^c_{r,u,m} t_{r,u,m} \le d, \forall c \in \mathcal{C} \tag{5}$$

where $x^c_{r,u,m}$ is a single binary variable equal to 1 if the data of user $u \in \mathcal{U}_r$ is coded using MCS $m \in \mathcal{M}$ and is processed on CPU core $c \in \mathcal{C}$. If not, it is equal to 0.
The objective function (1) maximizes the sum of overall users' throughput in the system or total system throughput. Note that the throughput values obtained with this objective may serve as a reference value to estimate the cost of fairness of the objective functions we will present next. MTT solution possesses the following constraints: (2) ensures that the decision variable $x^c_{r,u,m}$ only takes values 0 or 1. Equation (3) ensures that the data belonging to a given user $u \in \mathcal{U}_r$ are encoded using at most one MCS index m and are processed on at most one CPU core c. Equation (4) ensures that the decision-maker should not assign to any user an MCS index higher than the maximum allowed one because a higher MCS index would increase the decoding error to more than 10%, and finally (5) ensures that the data to be processed on core c have to finish before the deadline d. Intuitively, MTT favors users with high MCS indexes as they possess the higher throughput in the system and hence sacrifices users with lower MCS indexes.

2) *Admit All Users (AAU)*: Instead of privileging users with high throughput over others as done in MTT, AAU solution keeps the same objective function of maximizing the overall system's throughput while ensuring that all users have to be scheduled on the BBU pool. Compared to MTT, there is a slight modification in only one constraint: the single-core and MCS assignment constraint (3), while the objective function and other constraints remain the same as in MTT. This constraint can now be modified as follows:

$$\sum_{c\in\mathcal{C}}\sum_{m\in\mathcal{M}} x^c_{r,u,m} = 1,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{6}$$

It is worth mentioning that since AAU solution requires all users to be scheduled on the BBU pool, there is an upper bound on the number of users that can be admitted in the BBU pool depending on the capacity of the CPU cores in the BBU pool. When the BBU pool becomes highly overloaded, AAU solution schedules the users and assigns low MCS indexes to ensure all of them are admitted into the BBU pool. However, sometimes even the lowest MCS indexes are not enough to ensure the admission of all users, and in that case, AAU solution turns out to be infeasible.

3) *Maximize total Users' Satisfaction (MUS)*: Intuitively, the two previous solutions do not ensure a good level of fairness among the users. For that reason, we propose the MUS policy that aims to maximize the total users' satisfaction ratio and ensure a good level of fairness. We define the user satisfaction ratio as the ratio of the throughput achieved when the user operates using an adjusted MCS index, to the maximum throughput obtained when operating using the maximum allowed MCS index. The objective function of MUS solution is the following:

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}}\sum_{c\in\mathcal{C}} x^c_{r,u,m} \times \frac{b_{r,u,m}}{b_{r,u,max}} \tag{7}$$

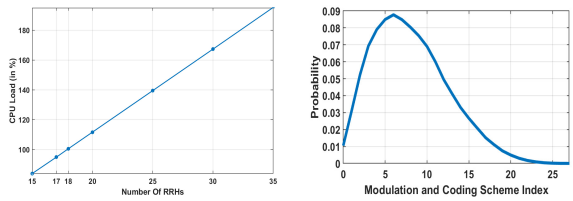
It ensures that when a given user $u \in \mathcal{U}_r$ has a maximum allowed MCS index $M_{r,u,max}$, the coordination entity

between radio and computing schedulers does not assign him an MCS index that deviates much from the maximum allowed MCS index. Hence, the user satisfaction ratio is maximized when the maximum allowed MCS index is used. We note that MUS solution maintains the same constraints as those of MTT.

IV. Performance Evaluation of ILP-based Coordination Policies

In this section, we present the simulation environment and the metrics we used to evaluate the performance of the proposed solutions.

We consider a BBU pool composed of 4 CPU cores that process the incoming data from the RRHs' users. We also vary the number of RRHs connected to the BBU pool from 15 to 35, which in turn varies the load of the BBU pool. Supposing



(a) BBU pool load as a function of RRHs' number (b) Probability distribution function of MCS indexes as in [6]

Fig. 3: CPU load and MCS Distribution

that each user operates with its maximum allowed MCS index, Fig. 3(a) provides the BBU Pool load as a function of the number of RRHs. The figure shows that the load varies from 83% to 195%. It is worth mentioning that the BBU pool starts to be fully loaded when the number of RRHs connected to the BBU pool is more than 17 RRHs. Intuitively, our priority is to focus on such a scenario because the case of a non-fully-loaded BBU pool allows all the users to operate using their maximum allowed MCS indexes. Hence, decreasing their MCS indexes is not beneficial at all. The RRHs operate using a 20 MHz bandwidth, so the number of available physical resource blocks (RBs) per TTI equals 100. These RBs are randomly assigned to the users connected to each RRH, and each user is allocated between 10 to 30 RBs. In order to use a real traffic distribution as a function of the MCS indexes, we consider the same probability distribution function as in [6] that is obtained using real measurements from [23]; this distribution is shown in Fig. 3(b), and we use it to sample the maximum allowed MCS indexes for the different users. Using this real-measurements-based distribution implicitly takes into account the interference control in the MCS allocation. Furthermore, to determine how much time is needed to process each user's data, we use the model from [9] that can be applied to single cell-user association scenarios[2], where Open Air Interface (OAI) RAN simulator is used. The formula is given as follows:

$$t_{r,u,m[\mu s]} = \frac{N_{RB}}{f^2_{[\text{GHz}]}} \sum_{i=0}^{2} \alpha_i m^i \tag{8}$$

- $t_{r,u,m}$: processing time of user $u \in \mathcal{U}_r$ using MCS index m.
- N_{RB}: The number of RBs used by user $u \in \mathcal{U}_r$.
- f: the clock frequency of the CPU
- m: the MCS index used by user $u \in \mathcal{U}_r$
- α_i: polynomial coefficients

According to [9], the values of alpha corresponding to the overall uplink processing time are: $\alpha_0 = 35.545$, $\alpha_1 = 1.623$, and $\alpha_2 = 0.086$. Arbitrarily, we set the CPU frequency to 4GHz. It is worth mentioning that the processing times strongly increase with the MCS index for a fixed number of RBs. However, we note that many more bytes are processed for larger MCS, and the processing time per byte decreases as the MCS index increases.

Moreover, each user's throughput is determined using the technical specification of ETSI [21]. The throughput of one user is determined by mapping its number of allocated resource blocks and its MCS index to the transport block size TBS (i.e., the data payload that can be carried by the physical layer). We note that the TBS of a user increases with the MCS index or the number of resource blocks. We get the throughput of each user by dividing its TBS by the transmission time interval (TTI) that is set to 1ms. Additionally, we use MATLAB to code and run the simulation, and we use CPLEX MILP solver interfaced with MATLAB to solve the ILP problems.

We compare the performance of the three proposed scheduling policies using different performance metrics, and we monitor the evolution of their performance as a function of the BBU pool load. The performance metrics used in this paper are the following:

- Average throughput: The average user throughput.
- The Number of admitted users: The number of users scheduled in the BBU pool and processed before the deadline.
- Fairness: We used the Jain's fairness index J_I [24] to compare the fairness of the three policies; it is given by:

$$J_I = \frac{\left(\sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{U}_r} s_{r,u}\right)^2}{\left(N \times \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{U}_r} s^2_{r,u}\right)} \tag{9}$$

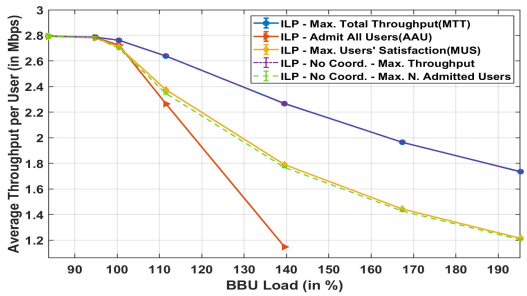
  For each user $u \in \mathcal{U}_r$, $s_{r,u}$ is its satisfaction ratio (i.e., the ratio of the attained throughput to the maximum achievable throughput achieved when using the maximum allowed MCS index). Also, N is the total number of users from all RRHs. A user is most satisfied if it gets the maximum throughput that can be achieved, i.e., being assigned its maximum allowed MCS index.
- Wasted power: This metric shows the ratio of the wasted power to the total emitted power. The power is useful when the data carried by the signals get processed before the deadline of 2ms. In contrast, data that is not processed before the deadline must be retransmitted. Hence the signal, and consequently its power, will be wasted.

In the following subsections, we evaluate the performance of the three different scheduling solutions concerning these four metrics. We limit the study and analysis to only one TTI instance, leaving the multiple instants scenario for future work. In addition, we compare our approaches to two other basic approaches in the literature [6] that do not consider any coordination between radio and computing schedulers. Their objectives are maximizing throughput and the number of Admitted Users, respectively. It is worth mentioning that 100 simulations were performed, and the confidence intervals of 95% are provided in the following results.

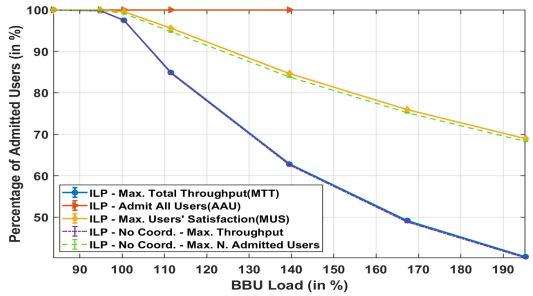
A. Average Throughput Per User

Fig. 4(a) shows the average throughput per user obtained by each proposed approach as a function of the BBU pool

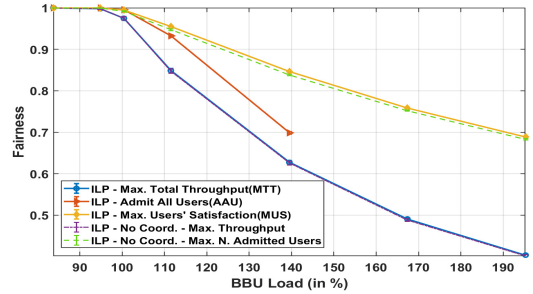
[2]For multi-cell association scenarios (i.e., where a user is served by multiple RRHs), a processing time model that takes into account other parameters has to be devised. These parameters should consider which RRHs are used for signaling and for traffic and how data is split among them.



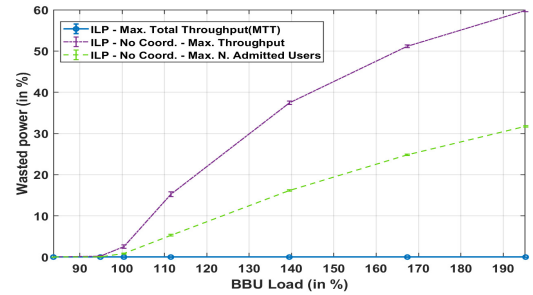
(a) Offered throughput (in %) as a function of BBU pool load



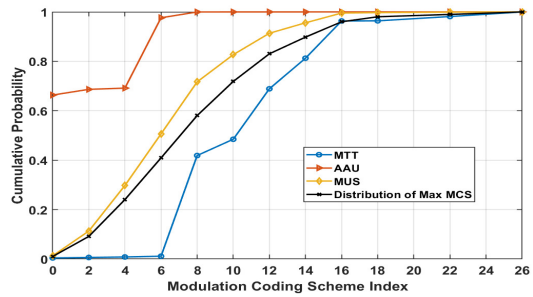
(b) Admitted Users (in %) as a function of BBU pool load



(c) Jain's Fairness Index as a function of BBU pool load



(d) Wasted Power (in %) as a function of BBU pool load



(e) Cumulative Distribution Function of the users' MCS indexes when the RRHs number is equal to 25

Fig. 4: Performance evaluation of the different scheduling solutions

load. We note that this throughput is normalized with respect to the total throughput's demand corresponding to the usage of the maximum MCS index by all users. The MTT solution clearly outperforms the other two coordination solutions in average throughput when the BBU pool load surpasses 100%. In contrast, when the BBU pool load is less than 100%, the different solutions achieve the same performance as the BBU pool can process all users' data while operating using their maximum allowed MCS indexes.

On the other hand, AAU shows the worst performance among the coordination policies with respect to the throughput metric because it requires the admission of all users from all RRHs to the BBU pool; hence the radio scheduler has to decrease the MCS indexes (which lowers the throughput) of many users so they may be scheduled on the BBU pool. Consequently, this severely degrades the total system throughput; solving AAU when the BBU load exceeds 140% is infeasible.

The MUS policy scores in-between results compared to the other policies. When comparing MTT and MUS policies to the non-coordination schemes, we find that MTT results almost coincide with Maximizing Throughput objective, and MUS results almost coincide with those found for Maximizing the Number of Admitted Users; the proposed coordination brings a slight improvement of less than 1%. The coordination policies present an advantage over the non-coordination schemes. Due to the allowed reassignment of MCS indexes, coordination schemes can use the CPU idle time to allocate more users and improve the system throughput. When it is impossible to use the maximum MCS for some users, it could be possible to use a lower MCS index making a place for one or more additional users. As can be seen in constraint (3), this extra degree of liberty compared to the non-coordination schemes (where a single MCS may be assigned) can only be beneficial, producing higher throughput and number of served users or at least as good as the no-coordination.

B. Number of Admitted Users

Fig. 4(b) shows the percentage of admitted users as a function of the BBU pool load; this percentage is relative to the total number of users from all RRHs. Intuitively, the AAU policy ensures the admission of all users (i.e., the percentage of admitted users is always 100%) as long as the problem is feasible, and it clearly outperforms all other solutions with respect to this metric. With respect to AAU, we see that the performance of MTT policy drops, admitting only 40.2% of users for an RRH number of 35 per BBU pool (i.e., equivalent to a BBU pool load of 195%). MTT tends to admit fewer users with higher throughput than to admit more users with low throughput. For the third policy (MUS), its performance gradually drops until it reaches 68.9% when the BBU pool load is 195%. Again, the performance of the third policy comes in-between that of the other two policies, as it aims to maximize fairness without necessarily admitting all users. In comparison with no-coordination schemes, we again notice a slight improvement, as explained in IV-A.

C. Fairness Index

The performance of the different policies is also measured with respect to the fairness in resource distribution, and we use for that purpose Jain's fairness index J_I as defined previously. We note that $J_I = 1$ is the maximum fairness value while $J_I = 0$ expresses the most unfair scenario. Here again in Fig. 4(c), for a load less than 100%, all users may use their

maximum MCS index resulting in a Jain index equal to 1 for all policies. However, when the number of RRHs per BBU pool increases, the fairness index declines.

The MUS policy outperforms all the other policies in terms of overall fairness as it maximizes users' satisfaction rate. On the contrary, the MTT policy is the least fair among the coordination policies. The reason is that it favors users who can achieve high throughput (i.e., those with high MCS indexes) and sacrifices those with lower MCS indexes that provide lower throughput. The AAU policy, with its objective of admitting the maximum number of users, for some time achieves the highest Jain index. Then beyond a certain load level, this objective results in an unfair share; adding more users to the system worsens the fairness index since many users would take a small portion of their maximum allowed throughput.

In comparison with no-coordination schemes, here again, we observe slight improvements, as explained earlier. The coordination schemes are slightly fairer compared to the no-coordination because they allow more users to get a chance to transmit by adjusting their MCS indexes to lower values. In contrast, the no-coordination schemes ignore these users since it is impossible to process their data if the maximum MCS is used. As a result, the coordination schemes are fairer than their no-coordination counterpart.

We recall that, when analyzing all the performance metrics, all the policies perform similarly when the BBU load is less than 100%. However, they behave differently when the BBU pool becomes fully loaded. When the BBU pool becomes overloaded, the different policies begin to adjust the MCS indexes of users since it is impossible to fit all users if they operate using their maximum MCS indexes. The selection of the users to be scheduled and their corresponding MCS indexes differentiates the coordination policies one from another.

D. Wasted Power

In Fig. 4(d), we plot the percentage of wasted power to the total emitted power. We define the wasted power as the power used to transmit user frames that the BBU pool will not process before the HARQ deadline due to the lack of processing resources and that will thus be retransmitted. In the coordination schemes, only users whose frames can be processed (eventually with a reduced MCS index) before the HARQ deadline would transmit data. Hence, they present a 0% waste of transmission power.

When we consider the no-coordination schemes, transmission decisions are taken by the radio scheduler alone without knowing whether the BBU pool will be able to process users' data. In this case, we notice a significant degradation of wasted transmission power. For the Maximizing throughput objective and Maximizing the number of admitted users objective, the wasted power increases until it reaches 59.8% and 31.7%, respectively, when the BBU load is 195%. Here we notice a significant benefit of the proposed coordination between radio and CPU scheduling: it saves considerable power.

E. MCS Selection Distribution

To better understand the strategy each policy follows to select the users to be scheduled and to assign their MCS indexes, we plot in Fig. 4(e) the cumulative distribution function of the selected MCS indexes when the number of RRHs per BBU pool is equal to 25, along with the curve of the maximum allowed MCS indexes. The latter distribution includes all users from all RRHs, whether they were admitted or not. The previous curves were only concerned with users who were admitted. The results show that the AAU policy, to ensure the admission of all the users in the BBU pool, forces the radio scheduler to enormously decrease the MCS indexes of users. In particular, the median value in the AAU policy is 0, meaning that 50% of users operate with the lowest MCS index, which is 0. However, no user under this policy uses an MCS index higher than 6. Looking at the MTT policy, we notice that it favors users with higher MCS indexes. The median of the corresponding CDF is equal to 10, which means that 50% of the users operate using an MCS index higher than 10. Moreover, the 90^{th} percentile for MTT is around the MCS 15, meaning that 10% of the users have an MCS index higher than 15. This behavior emphasizes that MTT's strategy is to schedule almost all the high-MCS users, leaving those with low MCS indexed with no resources. On the other hand, the CDF of the MUS policy is similar to that of the MAX MCS initially assigned to users. This justifies its fairness since the similarity of the distributions indicates that the MUS policy tries to assign each user an MCS close to the maximum allowed one; it attempts to make users fully satisfied as much as possible. With a probability greater than 0.6, the MUS policy will select an MCS between 4 and 12.

In conclusion, we can confirm that while the MTT policy favors the selection of high MCS index users and the AAU policy favors the selection of low MCS indexes, the MUS manages to strike a balance that minimizes the harm both on high and low throughput users. Moreover, we have shown that the proposed coordination scheme brings improvements, especially in reducing the amount of wasted power by up to 48%. This is an important finding, especially in the era of green RAN. Furthermore, with respect to the other metrics, the slight improvement of (1%-2%) could be of noticeable importance for operators, given the limited network resources. On the other hand, a practical implementation of the proposed coordination cannot be based on solving an optimization problem that may require heavy computational resources. It is necessary to propose low-complexity heuristics that can achieve a performance close to that of the ILP coordination solutions and allocate resources in real-time. In the following section, we discuss a few proposed heuristics.

V. Proposed Heuristics

In practice, mobile network operators should be able to dynamically allocate resources in a relatively short duration. While the proposed ILP coordination solutions manage to show enhancements over the non-coordination ones, it is not practical for the operator to solve an Integer Linear programming whenever it needs to allocate resources to users. Solving ILP problems requires a lot of computational resources. It could thus be computationally infeasible to solve them in real-time. It is necessary to switch our focus to low-complexity algorithms that utilize the coordination principle and can output sub-optimal MCS allocations in a short time. For this reason, we propose and evaluate three heuristics that can be used as alternatives to the ILP-based algorithms presented in section III-B.

A. The three proposed heuristics

In this section, we propose three heuristics that consider the adjustment of the MCS indexes of users. We refer to the parameters and variables presented in Table I:

- *Heuristic 1 - Prioritize High MCS*: Apply a two-level sorting to all users from all RRHs, firstly in descending order of maximum allowed MCS-Index and then in ascending order of maximum achievable throughput. An adjustment

Algorithm 1: Heuristic 1 Prioritize High MCS **&** Heuristic 3 Prioritize Low Throughput

input : $\mathcal{R}$, $\mathcal{U}_{r\in\mathcal{R}}$, $\{M_{r,u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\}$.
initialize:
1) Put all users $u \in \mathcal{U}_r$ from all RRHS $r \in \mathcal{R}$ in a list $\mathcal{L}$ and sort them according to the chosen heuristic;
2) $AdjMargin \leftarrow 0$;
3) $maxMCS \leftarrow \max(\{M_{r,u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$;
4) $AvTime(c) = d$, $\forall c \in \mathcal{C}$;
5) $x^c_{r,u,m} \leftarrow 0$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$;
while $AdjMargin \leq maxMCS$ **do**
  for $u \in \mathcal{L}$ **do**
    $m \leftarrow (M_{r,u,max} - AdjMargin)$;
    if $m >= 0$ **then**
      if $\exists c \in \mathcal{C}$ *such that* $t_{r,u,m} < AvTime(c)$ **then**
        $x^c_{r,u,m} \leftarrow 1$;
        Remove u from $\mathcal{L}$;
        $AvTime(c) \leftarrow (AvTime(c) - t_{r,u,m})$;
      end
    end
  end
  $AdjMargin \leftarrow AdjMargin + 1$;
end
output : $x^c_{r,u,m}$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$.

Algorithm 2: Heuristic 2 Admit All Users

input : $\mathcal{R}$, $\mathcal{U}_{r\in\mathcal{R}}$, $\{M_{r,u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\}$.
initialize:
1) Put all users $u \in \mathcal{U}_r$ from all RRHS $r \in \mathcal{R}$ in a list $\mathcal{L}$ and sort them accordingly as explained in the text;
2) $SysMCS \leftarrow 0$;
3) $maxMCS \leftarrow \max(\{M_{r,u,max} : u \in \mathcal{U}_r, r \in \mathcal{R}\})$;
4) $AvTime(c) \leftarrow d$ $\forall c \in \mathcal{C}$;
5) $x^c_{r,u,m} \leftarrow 0$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$;
6) $selectedMCS_{r,u} \leftarrow -1$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
7) $selectedCPU_{r,u} \leftarrow -1$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
8) $t_{r,u,-1} \leftarrow 0$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
9) $AvTime(-1) \leftarrow 0$;
while $SysMCS \leq maxMCS$ **do**
  for $u \in \mathcal{L}$ **do**
    $m \leftarrow \min(\{M_{r,u,max}, SysMCS\})$;
    if $m > selectedMCS_{r,u}$ **then**
      $AvTime(selectedCPU_{r,u}) \leftarrow (AvTime(selectedCPU_{r,u}) + t_{r,u,selectedMCS_{r,u}})$;
      if $\exists c \in \mathcal{C}$ *such that* $t_{r,u,m} < AvTime(c)$ **then**
        $selectedCPU_{r,u} \leftarrow c$;
        $selectedMCS_{r,u} \leftarrow m$;
      end
      $AvTime(selectedCPU_{r,u}) \leftarrow (AvTime(selectedCPU_{r,u}) - t_{r,u,selectedMCS_{r,u}})$;
    end
  end
  $SysMCS \leftarrow SysMCS + 1$;
end
$x^{selectedCPU_{r,u}}_{r,u,selectedMCS_{r,u}} \leftarrow 1, \forall r \in \mathcal{R}, u \in \mathcal{U}_r$;
output : $x^c_{r,u,m}$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$.

margin variable $AdjMargin$ is initialized to zero; this variable limits how much a user's MCS can deviate from the Maximum allowed MCS-Index. Then, the algorithm loops over the sorted users trying to admit them. After each complete loop, the algorithm increases the variable $AdjMargin$ by 1, then loops again over all sorted users. The algorithm stops when $AdjMargin$ becomes greater than $MaxMCS$ parameter; the latter is defined as the highest MCS index among all users. The detailed algorithm is presented in Algorithm 1.

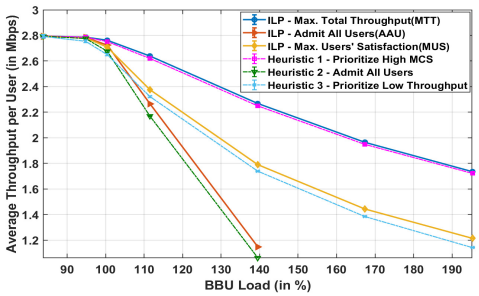
- *Heuristic 2 - Admit All Users*: Apply a two-level sorting to all users from all RRHs; firstly in ascending order of maximum MCS-Index; then in ascending order of maximum achievable throughput. A variable called $SysMCS$ is initialized to zero. This variable defines a limit on the MCS index that all users can use. Afterward, the algorithm loops over the sorted users. In each loop, the algorithm attempts to admit the users with the minimum of two indexes; $SysMCS$, and the maximum allowed MCS index of a user, $M_{r,u,max}$. Once the loop is completed, the $SysMCS$ is increased by one, and the users attempt to use the modified $SysMCS$ depending on the available computing resources. The algorithm terminates when $SysMCS$ exceeds the highest MCS index among all users, $MaxMCS$. The complete algorithm is presented in Algorithm 2.
- *Heuristic 3 - Prioritize Low Throughput*: The algorithm acts the same as in Heuristic 1 except in the sorting order; instead of applying a two-level sorting, all users are sorted in ascending order of maximum achievable throughput. Again, the detailed algorithm is presented in Algorithm 1.

B. Performance Analysis of the proposed heuristics

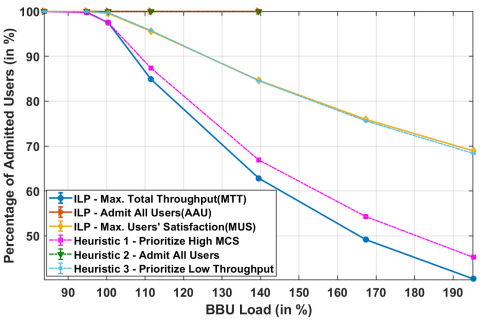
The proposed heuristics are evaluated and compared to the ILP-based policies of section III-B with respect to the same metrics of section IV: Average Throughput per User, Number of Admitted Users, and Fairness Index. The results are depicted in figures 5(a), 5(b), and 5(c), respectively. We note from the figures that:

- *Heuristic 1 - Prioritize High MCS* shows very close results to those obtained by the first ILP problem, MTT, especially concerning the throughput metric because it prioritizes users with high MCS indexes. However, it outperforms MTT with respect to the other two metrics: the number of admitted users and the fairness metrics. Compared to MTT, this heuristic can score up to a 4.5% improvement concerning the percentage of admitted users metric and up to 0.049 of improvement for the fairness metric.
- *Heuristic 2 - Admits All Users* aims to admit all users in the system; hence its performance regarding the percentage of admitted users is the same as AAU policy. It deviates slightly from AAU policy with respect to the throughput metric and can worsen the performance with a maximum drop of 4%. With respect to the fairness index metric, this heuristic and AAU can deviate from each other with a difference not larger than 0.06. As mentioned earlier, AAU is no longer feasible once the BBU pool exceeds 140%.
- *Heuristic 3 - Prioritize Low Throughput* has more or less a similar performance compared to MUS policy concerning all metrics. Concerning the throughput metric, the performance of this heuristic would drop by up to 6%. The difference between the ILP and the heuristic is very slight for the metrics of admitted users and fairness.

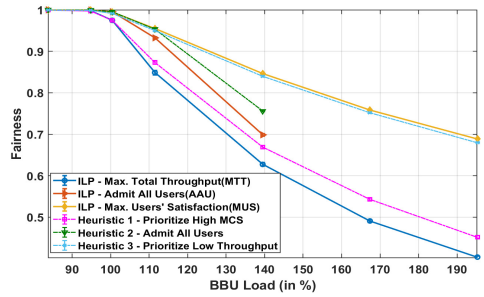
Compared with its corresponding ILP counterpart, each of the three heuristics scored close results concerning all performance metrics. We recall that the three heuristics score 0% concerning the metric of wasted power. Like the ILP



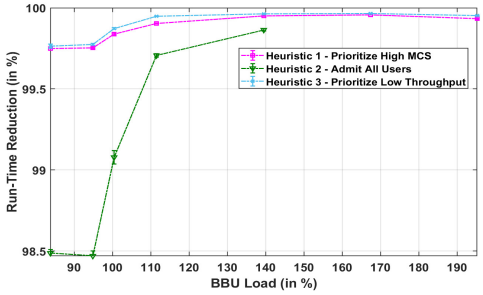
(a) Offered throughput (in %) as a function of BBU pool load



(b) Admitted Users (in %) as a function of BBU pool load



(c) Jain's Fairness Index as a function of BBU pool load



(d) Reduction of Elapsed Time of each heuristic as a function of the number of RRHs with respect to its ILP algorithm

Fig. 5: Comparison of the performance of the heuristics in comparison to ILP problems

coordination solutions, users aware that the BBU can not process their data will not transmit, thus saving the transmission power. In short, the proposed heuristics can serve as practical replacements for the high-complexity ILP algorithms and can be implemented by the mobile operators to apply real-time scheduling.

C. Computational Complexity

The main motive behind proposing the heuristics is the high computational complexity of ILP solvers. Finding real-time scheduling results is essential for mobile operators; otherwise, our proposal will not have practical grounds and will remain theoretical.

For both Algorithm 1 and Algorithm 2, the sorting part should, in the worst case, take no more than $|\mathcal{R}|^2 \cdot |\max_{r\in\mathcal{R}} \mathcal{U}_r|^2$, supposing that in each iteration, each user will be compared with all other users. For the second part of each algorithm, the worst-case scenario corresponds to iterating over all MCS indexes, and for each MCS index, iterating over all users. Hence, the number of iterations for the second part of each algorithm is $|\mathcal{R}| \cdot |\max_{r\in\mathcal{R}} \mathcal{U}_r| \cdot |\mathcal{M}|$.

Experimentally, the proposed heuristics can find solutions much faster than the corresponding ILP algorithms. In figure 5(d), we plot the graphs of the percentage of reduction of Elapsed Time of each heuristic as a function of the number of RRHs with respect to its corresponding ILP algorithm. We have made the study using a computer running on Intel® Core™ i9-9880H Processor, and the ILP solver is CPLEX for MATLAB. While Heuristic 2 has achieved more than 98.6% reduction in run-time with respect to AAU, Heuristics 1 and 3 achieved more than 99.7% reduction with respect to MTT and MUS, respectively. The achieved reduction in run-time is very significant.

VI. Multi-Services Scenario

The fifth-generation (5G) New Radio (NR) of mobile communications has been designed to support two major classes of services with vastly heterogeneous requirements: ultra-reliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB) [8]. On the one hand, eMBB supports stable connections with very high peak data rates and moderate rates for cell-edge users. On the other hand, URLLC supports low-latency transmissions of small payloads with very high reliability from a limited set of terminals.

Few approaches have been adopted in the Third Generation Partnership Project (3GPP) standard [25] to handle the coexistence of these two services. One possible way is to slice the radio resources and reserve a portion for URLLC traffic [8]. Another approach is multiplexing eMBB and URLLC on shared radio resources while prioritizing the latter [8]. The latter can puncture ongoing eMBB transmissions and transmit instead of them. URLLC transmission can happen at the start of an sTTI (Short Transmission Time Interval). Hence their transmission time is much shorter than eMBB transmission time.

So far, we have demonstrated the benefits of the proposed coordination and proposed low-complexity heuristics, alternatives to the ILP-based coordination algorithms. The scenarios we tested in the previous sections go under the eMBB category, and we have not considered the effect of URLLC service transmission subject to tight latency constraints. While the coordination policies achieve better results than no-coordination for eMBB traffic, it is interesting to study the impact of URLLC traffic on the performance of our proposed coordination policies. The impact of URLLC frames exists when the computing power is shared by both resources, as it is the case in our study. In particular, the incoming URLLC traffic during the processing period of eMBB transmissions cannot be delayed due to the strict latency requirement. The coordination does not modify the MCS index of URLLC transmission but only controls the MCS index of eMBB frames. However, the URLLC frames are prioritized over eMBB frames. In other words, once URLLC frames arrive, the BBU pool should process them and preempt eMBB frames.

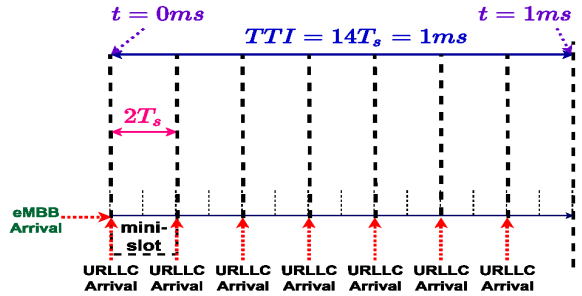


Fig. 6: eMBB and URLLC frames arriving in the same TTI share a set of computing resources. T_s is the duration of one OFDM symbol.

Given that URLLC frames arrive at the BBU pool every sTTI and are prioritized over eMBB transmission, the performance of the heuristics may change depending on the arrival rate of URLLC frames. Hence it remains important to analyze how the heuristics perform for different URLLC frames arrival rates. It is worth mentioning that on the radio level, it is possible that either there are radio resources reserved for URLLC transmissions or that URLLC transmission punctures ongoing eMBB transmissions and transmits instead. However, in our paper, regardless of how URLLC traffic was transmitted at the radio level, we only focus on the effect of the competition for computing resources on the coordination performance, and we leave the effect of puncturing on the radio level for future work.

A. Scenario Description

Targeting reliable and low latency communications, the 5G-REL15 [25] [26] introduced mini-slots; the time required to transmit a transport block over this mini-slot is equal to the sTTI length. Hence, URLLC traffic arrives at the BBU pool every sTTI duration. Enabling the mini-slots option paves the way for URLLC users to achieve their low latency requirement, although this may come at the expense of eMBB users. As URLLC users have higher priority, they need to be processed, even if this leaves eMBB frames unprocessed before the 2ms deadline. Figure 6 shows the scenario under study. We suppose that eMBB frames and URLLC frames that arrive during the same TTI will be processed by the same set of CPUs so that the processing of eMBB frames from different TTIs will not happen on the same CPUs to avoid overlapping. However, we limit the study to only one TTI, so only one set of CPUs is needed. One TTI consists of 14 OFDM symbols. In addition, we consider the underlying short TTIs (sTTIs) to be 2-OFDM symbols long. At each sTTI, URLLC frames arrive, and the CPUs should process them while preempting the ongoing processing of eMBB frames.

We consider the same scenario described in Section IV, but with the existence of URLLC or eMBB users. We suppose that the arrival of URLLC packets in each mini-slot follows a Poisson process with an arrival rate λ (in our simulation, λ varies between 0 and 5 users per RRH per sTTI). We note that for $\lambda = 0$, we get the scenario with eMBB users only, while $\lambda = 5$ represents a very extreme case of an average URLLC frames arrival of 35000 frames per second. To estimate the processing load of URLLC packets on the BBU pool, we additionally need to determine their processing time. Referring to equation (8), this processing time depends on the number of RBs used by these frames. The MCS used by URLLC users is sampled as in section IV. For the required number of RBs to be used by URLLC users, we use the following formula [19]:

$$N_{r,u}^{LC} = \frac{PacketLength}{Q_m \times N_{MiniSlot}^{RE} \times N_{RB}^{SC} \times CodeRate \times OH} \quad (10)$$

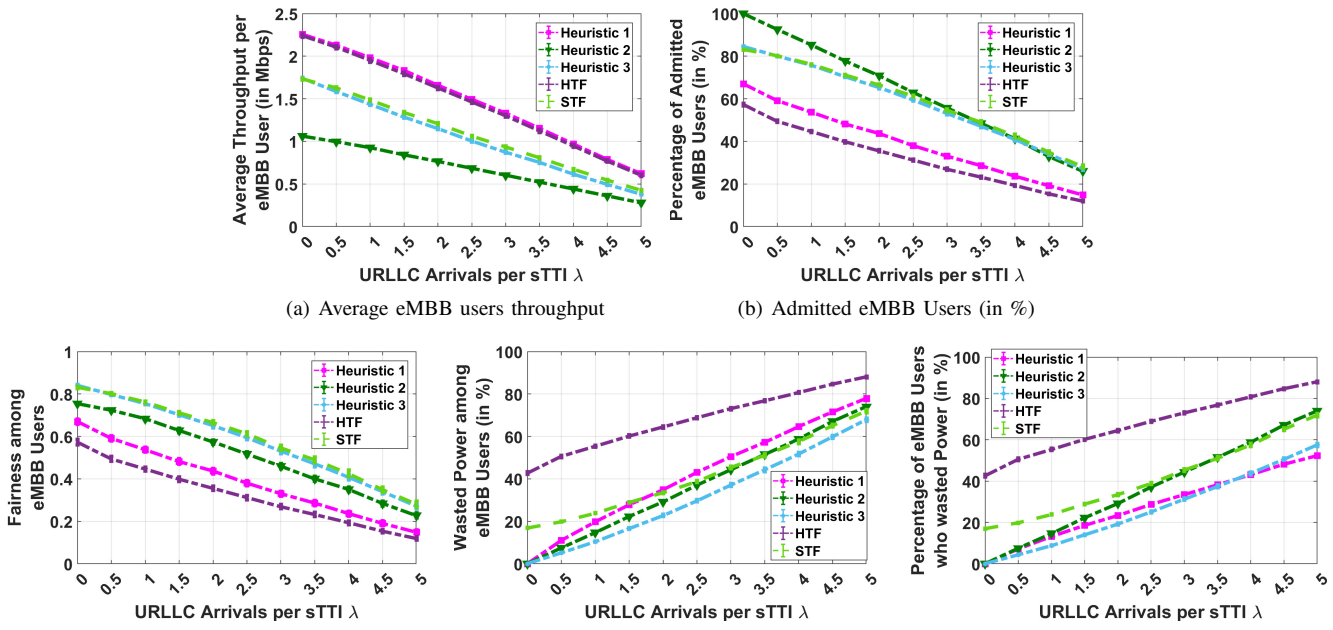
Q_m is the number of bits per symbol. Q_m and the $CodeRate$ are determined from the tables in [22]. $N_{MiniSlot}^{RE}$ is the number of Resource Elements (OFDM symbols) per a mini-slot per 1 sub-carrier (i.e., 2, 4, or 7), N_{RB}^{SC} is the number of sub-carriers per a Resource Block (i.e., 12), and OH is the overhead. Similar to [20], we suppose that the length of URLLC packets, $PacketLength$ is 32 bytes. As in [19], OH is equal to 0.715. For the sake of benchmarking, we consider two additional No-Coordination heuristics from [6]: High Throughput First (HTF), which prioritizes users with the highest throughput, and Short Time First (STF), which prioritizes users with the shortest processing time. In these algorithms, users transmit with their max MCS, and the BBU pool can only decide to process users or dismiss them. The simulation is run 100 times, and the 95% confidence intervals are shown. We note that the confidence interval may appear very tight in the figures. All the other parameters in the simulation environment described in IV are maintained.

B. Performance evaluation of the heuristics in a multi-services environment

In the following, we evaluate the performance of our proposed heuristics in Section V with the coexistence of eMBB and URLLC users, using the metrics of throughput, Admitted Users, Fairness Index, and wasted power. We note that these metrics depict the performance of eMBB users only; because URLLC users cannot adjust their MCS indexes, the coordination is irrelevant to them.

Fig. 7(a), 7(b), 7(c), 7(d), and 7(e) show the performance of the coordination heuristics as a function of URLLC users' arrival rate, concerning the metrics of eMBB average throughput per user, the total number of admitted eMBB users, Jain's Fairness Index, and the wasted transmission power. Fixing the number of RRHs to 25, we monitor the effect of URLLC users' arrival on the performance of eMBB users.

Concerning all metrics, the performance of the heuristics degrades as the URLLC arrival rate increases. As before, heuristic 1 achieves the highest average throughput among eMBB users in comparison to other heuristics. Additionally, while heuristic 3 achieves higher throughput than heuristic 2 for low URLLC arrival rate, as explained in previous sections, the performance of the two heuristics converges at the end. The reason is that these two heuristics process users based on the admission order. Heuristic 2 sorts users in the ascending order of the MCS index, then in the ascending order of throughput, while heuristic 3 sorts users in the ascending order of throughput. This increases the tendency to process users with low frame sizes first and keep users with high frame sizes until the end. On the other hand, heuristic 2 tends to admit potentially high throughput users with a more reduced MCS index to accommodate more low throughput users, similar to the behavior of ILP-AAU in Fig. 3(b). This justifies why heuristic 3 achieves higher throughput at low arrival rates. When URLLC frames arrive, they cause the CPU to preempt eMBB users' processing and start processing the URLLC frames that arrived. This would make it impossible for the eMBB users with longer frames, and thus higher throughput, to be processed before the deadline. At a very high URLLC arrival rate, most users with high and medium frame sizes fail to be processed before the 2ms deadline; the BBU pool would only process eMBB users with low MCS and throughput.



(a) Average eMBB users throughput (b) Admitted eMBB Users (in %)

(c) Fairness among eMBB users (d) Wasted Power of eMBB transmissions (in %) (e) eMBB users who wasted power (in %)

Fig. 7: Performance evaluation of our proposed heuristics as a function of URLLC users' arrival rate when the number of RRHs in the BBU pool is 25

As figure 7(b) shows, heuristic 2 can no longer admit all eMBB users in the presence of URLLC frames. Since the BBU pool is already fully loaded for a number of RRHs equal to 25, the computing resources are insufficient to process all eMBB and URLLC frames in 2 ms. Hence the arrival of URLLC frames leads to failure to process eMBB users. This violates the heuristic's primary goal, which is to admit all eMBB users. For the same reasons explained above, figure 7(c) shows the performances of heuristics 2 and 3 with respect to the fairness metric converge at higher URLLC arrival rates.

Analyzing Fig. 7(d), the heuristics employing coordination can no longer achieve 0 wasted transmission power. Users are initially promised to be admitted by the BBU pool. However, an increased arrival rate of URLLC frames leads to increased wasted transmission power. It is good to note that at a high URLLC arrival rate, selecting heuristic 2 becomes a bad idea compared to heuristic 3. The latter becomes a better choice considering all the metrics.

When comparing the coordination heuristics to the no coordination heuristics, HTF and STF, the existence of URLLC traffic reduces and diminishes the already slight improvement (1%-2%, as section IV shows) that the coordination brings concerning the metrics of throughput, admitted users, and fairness. The reason is that URLLC traffic negatively affects the users who managed to be admitted at the BBU pool by reducing their MCS index. Concerning the metric of wasted power, the coordination heuristics 1 and 3 remain better than the no-coordination counterparts, HTF and STF, respectively, because regardless of URLLC traffic, eMBB users, whom the BBU pool is initially unable to process, will not transmit and thus save transmission power. However, since the metric of wasted power is normalized with respect to the total transmission power and the coordination decreases the total transmission power, it would delude us to think that STF is better than heuristic 2. Hence, we plot the graphs of the percentage of eMBB users who wasted their transmission power with respect to the total number of users in the BBU pool in fig. 7(e). It is clear that the percentage of users who wasted their power for heuristics 1 and 3 is much better than the no-coordination heuristics. However, since heuristic 2 admits all users and then removes a lot of them to prioritize URLLC, it wastes more power than STF at $\lambda = 5$.

In short, even in the extreme case considered in this section, where random URLLC frames arrive with no predetermined processing resources in the BBU pool, we show that coordination heuristics between the radio MCS assignment and the BBU resources is effective in saving radio power and reducing the percentage of unprocessed eMBB frames. Among the heuristics presented, heuristic 3 seems to be more robust and offers the best performance in this context.

VII. Conclusion

In this paper, we investigated three ILP policies that implement coordination between radio and computing schedulers in Cloud-RAN context. Motivated by the fact that the data processing time strongly depends on the transmission MCS index, the coordination policies allow the radio scheduler to set the MCS index for users' transmission not only based on the radio conditions but also on the ability of the BBU pool to process users' data. The three coordination schemes (namely MTT, AAU and MUS) aim to maximize total throughput, admit all users, and maximize users' satisfaction. We have evaluated them according to different performance metrics. Results show that the proposed coordination achieves a vital improvement by significantly reducing the amount of wasted transmission power and bringing a slight but systematic improvement to the other metrics. Among the ILP coordination policies, the MUS policy is the fairest; it achieves in-between values of throughput and the number of allocated users in comparison with the other coordination policies. In addition, we proposed three low-complexity heuristics and compared their performance to that of the high-complexity ILP algorithms. We proved that the heuristics are good candidates to replace the ILP algorithms to achieve real-time performance. Moreover, we analyzed the performance of the heuristics in a multi-service environment, where users of different services (i.e., eMBB and URLLC) coexist. The results show that the

heuristics employing coordination can no more avoid wasting transmission power when URLLC traffic exists. However, they still reduce power consumption in comparison with no-coordination heuristics. For future work, we aim to extend our study and evaluate the coordination performance at the MAC layer, taking into account the transmission errors and subframe retransmission at the MAC-layer level. Moreover, we aim to consider dynamic RB allocation, power allocation, and interference control to study the benefits of joint radio and computing resource allocation. We also consider studying a scheme where the MCS selection is limited by the capacity of the fronthaul links. Furthermore, we plan to extend our study to multi-cell association scenarios considering a suitable processing time model.

REFERENCES

[1] M. Sharara, S. Hoteit, P. Brown, and V. Vèque, "Coordination between Radio and Computing Schedulers in Cloud-RAN," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 37–44.

[2] C. Mobile, "C-RAN: the road towards green RAN," *White Paper, ver*, vol. 2, pp. 1–10, 2011.

[3] O. Chabbouh, S. B. Rejeb, Z. Choukair, and N. Agoulmine, "A novel cloud RAN architecture for 5G HetNets and QoS evaluation," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[4] M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, "A comprehensive survey of RAN architectures toward 5g mobile communication system," *IEEE Access*, vol. 7, pp. 70 371–70 421, 2019.

[5] H. Khedher, S. Hoteit, P. Brown, R. Krishnaswamy, W. Diego, and V. Veque, "Processing time evaluation and prediction in Cloud-RAN," in *IEEE International Conference on Communications (ICC)*, 2019.

[6] H. Khedher, S. Hoteit, P. Brown, V. Véque, R. Krishnaswamy, W. Diego, and M. Hadji, "Real traffic-aware scheduling of computing resources in cloud-ran," in *International Conference on Computing, Networking and Communications (ICNC)*, 2020.

[7] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 5th ed. Springer Publishing Company, Incorporated, 2012.

[8] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[9] S. Khatibi, K. Shah, and M. Roshdi, "Modelling of Computational Resources for 5G RAN," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018, pp. 1–5.

[10] V. Q. Rodriguez and F. Guillemin, "Towards the deployment of a fully centralized Cloud-RAN architecture," in *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017.

[11] K. Boulos, K. Khawam, M. El Helou, M. Ibrahim, H. Sawaya, and S. Martin, "An efficient scheme for BBU-RRH association in C-RAN architecture for joint power saving and re-association optimization," in *IEEE International Conference on Cloud Networking (CloudNet)*, 2018.

[12] H. Taleb, M. El Helou, K. Khawam, S. Lahoud, and S. Martin, "Centralized and distributed RRH clustering in Cloud Radio Access Networks," in *IEEE Symposium on Computers and Communications (ISCC)*, 2017.

[13] Y. Li, H. Xia, J. Shi, and S. Wu, "Joint optimization of computing and radio resource for cooperative transmission in C-RAN," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2017.

[14] E. Aqeeli, A. Moubayed, and A. Shami, "Power-Aware Optimized RRH to BBU Allocation in C-RAN," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1311–1322, 2018.

[15] M. Elhattab, M. Kamel, and W. Hamouda, "Edge-Aware Remote Radio Heads Cooperation for Interference Mitigation in Heterogeneous C-RAN," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 142–12 157, 2021.

[16] M. M. Abdelhakam, M. M. Elmesalawy, M. K. Elhattab, and H. H. Esmat, "Energy-efficient BBU pool virtualisation for C-RAN with quality of service guarantees," *IET Communications*, vol. 14, no. 1, pp. 11–20, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-com.2019.0187

[17] L. Ferdouse, A. Anpalagan, and S. Erkucuk, "Joint communication and computing resource allocation in 5G cloud radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9122–9135, Sep. 2019.

[18] Y. Jia, H. Tian, S. Fan, P. Zhao, and K. Zhao, "Bankruptcy game based resource allocation algorithm for 5G Cloud-RAN slicing," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[19] D. Maaz, A. Galindo-Serrano, and S. E. Elayoubi, "URLLC User Plane Latency Performance in New Radio," in *2018 25th International Conference on Telecommunications (ICT)*, 2018, pp. 225–229.

[20] M. Elsayed and M. Erol-Kantarci, "Reinforcement Learning-Based Joint Power and Resource Allocation for URLLC in 5G," in *2019 IEEE Global Communications Conference (GLOBECOM)*.

[21] (2014) LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 12.3.0 Release 12).

[22] (2018) 5G; NR; Physical layer procedures for data, ETSI TS 138 214 V15.3.0.

[23] H. D. Trinh, N. Bui, J. Widmer, L. Giupponi, and P. Dini, "Analysis and modeling of mobile traffic using real traces," in *International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[24] R. Jain, D. Chiu, and W. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems*. DEC Research Report TR-301, Sep 1984.

[25] ETSI TS 136 213 V12.3.0, 3GPP, "Technical specification group services and system aspects; release 15 description," Tech. Rep., TR 21.915, v1.1.0, March 2019.

[26] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-Reliable and Low-Latency Communications in 5G Downlink: Physical Layer Aspects," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, 2018.

Mahdi Sharara received the diploma degree in electrical, electronics, computer, and telecommunications engineering from Lebanese University, Beirut, Lebanon, in 2018, and the M.S. degree in telecom and network from the Lebanese University and Saint-Joseph University in 2018. Currently, he is a PhD student at Université Paris-Saclay. His research interests include Resource Allocation in Mobile Networks, Cloud-RAN, and Open-RAN, along with Machine Learning and Game-Theoretic techniques.

Sahar Hoteit is currently an associate professor at Paris Saclay University/CentraleSupélec, France. She received the M.S.and PhD degree in network and computer science from the University of Pierre and Marie Curie (now Sorbonne University). Her research interests cover mobile networking, Internet of Things, game theory, Open-RAN and Cloud-RAN architectures.

Patrick Brown received an engineering degree from L'École Nationale Supérieure des Télécommunications de Paris in 1980 and a PhD in computer science from the University Nice and Sophia Antipolis, France, in 2005. He joined Orange Labs as a research engineer in 1982. His research interests include performance evaluation, modelling, and resource allocation for mobile communication systems and distributed applications.

Véronique Vèque obtained her PhD degree in communication networks in 1989 from University Pierre et Marie Curie - France. In 1990, she was an Associate Professor at University of Paris-Sud (Paris 11), and in 2000 to present, she worked as a full Professor at University of Paris-Sud/University Paris-Saclay. She is currently a research member of Laboratory of Signals and Systems. Her research interests lie in the field of both wireless, mobile networks, resource allocation and quality of service techniques.