Coordination between Radio and Computing Schedulers in Cloud-RAN

Mahdi Sharara, Sahar Hoteit, Patrick Brown, Véronique Vèque

Coordination between Radio and Computing Schedulers in Cloud-RAN

Mahdi Sharara*, Sahar Hoteit*, Patrick Brown and Véronique Vèque*
*Université Paris Saclay-CNRS-CentraleSupélec,
Laboratoire des Signaux et Systèmes, 91190, Gif-sur-Yvette, France
Emails: mahdi.sharara@universite-paris-saclay.fr, sahar.hoteit@universite-paris-saclay.fr,
brown.patrick2@gmail.com, veronique.veque@universite-paris-saclay.fr

***Abstract*—Cloud Radio Access Network is a promising mobile network architecture. It is based on decoupling Base Band Units (BBUs) from Radio Remote Units (RRUs) and on centralizing the baseband processing of many Radio Remote Heads (RRHs) in a BBU pool. Such architecture paves the way for many improvements in mobile networks such as increased flexibility, increased energy efficiency, reduced operational costs, and better user experience. However, as computing resources are shared among the RRHs connected to the BBU pool, it becomes increasingly 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. In fact, the amount of time required to process users' data is heavily dependent on radio parameters such as Modulation Coding Scheme index (MCS). In this paper, we propose to enable the coordination between radio and computing resources schedulers; such coordination allows the radio scheduler to adjust users' MCS indexes (i.e., which in turns adjusts the processing time), so that the computing resources scheduler can also schedule the processing of users' data on the BBU pool while respecting the HARQ-deadline. We propose three Integer Linear Programming (ILP)-based schemes and three low-complexity heuristics, and we evaluate the performance with respect to different performance metrics. The simulation results reveal the benefits of coordination solutions; especially, in terms of reducing the wasted transmission power. They also indicate for each metric the best coordination scheme that can be adopted.**

I. Introduction

Cloud Radio Access Network (C-RAN) is a key pillar in future Mobile Networks and consists in decoupling Base Band Units (BBUs) from Radio Remote Units (RRUs), and centralizing the baseband processing of many Radio Remote Heads (RRHs) in a shared BBU pool [1]. The latter processes some virtualized functions such as fast Fourier transform, demodulation, decoding, etc. Decoupling baseband processing from radio elements in C-RAN leads to multiple advantages as it reduces CAPEX and OPEX of network operators, eases the implementation of interference management mechanisms, increases flexibility and energy efficiency, and hence improves user experience [2].

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 massive number of RRHs to make sure it maintains the ability to processes users' data before passing the deadline imposed by the MAC-layer mechanism (Hybrid Automatic Repeat Request (HARQ)). Knowing that the processing times of users' data strongly depends on radio parameters (e.g., the Signal to Interference plus Noise Ratio (SINR), the Modulation Coding Scheme index (MCS) which permits to set the users' throughput by specifying the modulation and the code rate to be used) [3] [4], the coordination between both radio and computing schedulers raises as a candidate to efficiently manage the resources of an overloaded BBU pool. As a matter of fact, coordination solutions should be able to achieve the required quality of service for users and to ensure a good level of fairness.

Throughout this work, we propose different schemes that implement coordination between radio and computing schedulers in Cloud-RAN. This coordination permits the adjustment of users' MCS indexes in order to ensure the processing of their data on the BBU pool. In fact, the processing time of data increases as the MCS index increases [4]. Thus, when the BBU pool gets overloaded, it will not be able to process all users' data while respecting the HARQ-deadlines requirements. Users of non-processed data should re-transmit the data, as part of HARQ mechanism, and such retransmission turns out to be energy-inefficient phenomenon that reduces network performance, and wastes radio resources. Hence, instead of having to re-transmit the data, it could be more efficient to reduce data processing times by decreasing the corresponding MCS indexes of users. While decreasing 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, our aim is to study and evaluate some coordination schemes that allow the radio scheduler to assign users' MCS indexes not only based on the radio quality; the MCS assignment should take into account the availability of computing resources in the BBU pool, and whether they are sufficient to process users' data given the deadlines constraints. We note that the radio scheduler can only adjust users' MCS indexes to values lower than the maximum allowed indexes; the latter are computed according to the radio conditions (i.e., a user cannot use an MCS index higher than the maximum allowed one. Otherwise, transmission error increases, and may lead 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,

978-3-903176-32-4

is based on the adopted strategy. We compare and evaluate their performance according to different metrics such as total throughput, number of users that are admitted, fairness, and wasted power. Knowing that the complexity of ILP problems is NP-Hard, we propose three low-complexity heuristics that depict the performance of the ILP solutions. The results give insights to network operators on what policy to select depending on the metric they prioritize.

The rest of this paper is organized as follows: Section II presents the state of the art. In section III, we present the proposed coordination solutions, while section IV presents the performance evaluation of these solutions. Finally, we conclude our work in section VI.

II. Related Work

Cloud-RAN continues to receive a lot of interest in many research works. In [3], the authors study the different processing times of BBU up-link 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. Authors in [5], 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 in reducing the run time of the decoding function. In [4], the authors propose two different computing scheduling algorithms to schedule the processing of RRHs' data arriving every transmission time interval (TTI). These algorithms aim at increasing the number of correctly decoded sub-frames and the system throughput, respectively. The authors evaluate the performance of these algorithms as a function of the number of RRHs that are assigned to the BBU pool. Additionally, they compare their proposed algorithms to known scheduling heuristics which only focus on computing scheduling without taking into consideration the radio scheduling.

The authors in [6] and [7] consider the problem of RRH-BBU association and model it 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 [8], the authors consider the issue of joint radio and computing resources allocation in Cloud-RAN. They develop a two-independent steps approach which firstly works on the allocation of computing resources by mapping users to virtual machines, where each virtual machine is a BBU. Then, the radio resource allocation is carried out by controlling the beamforming vector design and the power for each user. In the same context, the authors in [9] 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. The proposed resource allocation solutions in the literature are different from our approach in this paper. To the best of our knowledge, we are the first to propose a coordination between radio and computing scheduler based on permitting the adjustments of users' MCS index.

III. Context and problem formulation

The system under study consists of a set of RRHs connected to a centralized BBU pool which is composed of homogeneous CPU cores with the same execution speed. As the uplink processing time is at least 7 times larger than that in downlink [3], 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 among each other the available resource blocks which 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 acknowledgement should be delivered to users in $8ms$ [5].

We further consider that the MCS index of a user is determined by jointly considering the channel conditions of all the RBs in the associated RRH. This permits the radio scheduler to attribute the same modulation and coding scheme (MCS) index to a given user over all its allocated resource blocks. It is worth mentioning that a maximum allowed MCS index is attributed by the radio scheduler to a given user by considering its radio conditions measured by 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), and it carries information on how good/bad the communication channel quality is [3]. Based on this indicator, the radio scheduler determines for each user, its maximum allowed Modulation Coding Scheme (MCS) index that can be used. As shown in [3], 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 MCS index. Hence, if the BBU pool is overloaded, and if all users use their maximum allowed MCS index, the BBU pool will fail to process all the incoming users' data in 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 re-transmit 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 used 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 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 was initially chosen by the radio scheduler for data transmission $M_{r,u,max}$. Based on the chosen index m, user u transmits an amount of data that is equal to $b_{r,u,m}$; the latter is determined according to [10] 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 simulation

[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 acknowledgement 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 (muti-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
$x^c_{r,u,m}$	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	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 $c \in \mathcal{C}$

results done in [3]. We suppose that each user transmits its data with a constant power P_t. 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 centralized single decision-maker.

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

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

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

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

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

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

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. 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) takes into consideration that the decision maker should not assign any user an MCS index that is higher than its maximum allowed one, 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 that is 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 the lowest MCS indexes to them 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.

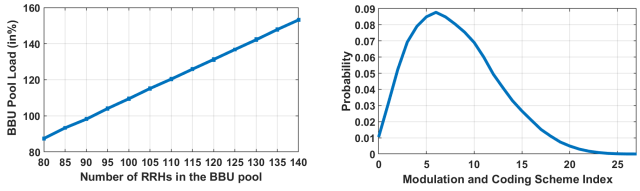
3) *Maximize total Users' Satisfaction (MUS)*: Intuitively, the two previous solutions do not ensure a good level of fairness among the users. For that reason, we propose the MUS policy that aims at maximizing the total users' satisfaction ratio and hence ensures a good level of fairness. We define the user satisfaction ratio by 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 exactly the same constraints as those of MTT.

IV. Performance Evaluation

In this section, we present the simulation environment along with the metrics we used to evaluate the performance of the



(a) BBU pool load as a function of RRHs' number (b) Probability distribution function of MCS indexes as in [4]

Figure 1: CPU load and MCS Distribution

proposed solutions. We consider a BBU pool composed of 4 CPU cores which has to process the incoming data from the RRHs' users. Also, we vary the number of RRHs connected to the BBU pool from 80 to 140 which in turn varies the load of the BBU pool.

Supposing that each user operates with its maximum allowed MCS index, Fig. 1(a) provides the BBU Pool load as a function of the number of RRHs. As shown in the figure, the load varies from 87% to 153%. 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 90 RRHs. Intuitively, our priority is to focus on such a scenario because the case of non fully-loaded BBU pool allows all the users to operate using their maximum allowed MCS indexes and 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 is equal to 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 [4] that is obtained using real measurements from [11]; this distribution is shown in Fig. 1(b) and we use it to sample the maximum allowed MCS indexes for the different users. Furthermore, in order to determine how much time is needed to process each user's data, we rely on the simulation results in our previous work in [3] where the Open Air Interface (OAI) RAN simulator is used. It is worth mentioning that the processing times found in [3], for a given number of RBs, strongly increases with the MCS index. However, note that many more bytes are processed for larger MCS and in fact the processing time per byte decreases as the MCS index increases. Moreover, the throughput of each user is determined using the technical specification of ETSI [10]. As a matter of fact, 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 increase of either the MCS index or the number of resource blocks or with the increase of both of them. We get the throughput of each user by dividing its TBS by the transmission time interval (TTI) that is set to 1ms. Moreover, 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:

- Total throughput: The overall users' throughput.
- 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 [12] to compare the fairness distribution of the three policies; it is given by: $J_I = \left(\sum_{m\in\mathcal{N}} s_i\right)^2 / (N \times \sum_{m\in\mathcal{N}} s_i^2)$, where s_i is the satisfaction ratio of each user (i.e., the ratio of the attained throughput to the maximum achievable throughput achieved when using the maximum allowed MCS index as defined in section III-B), $\mathcal{N}$ is the set of users, and N the number of users. 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 re-transmitted. Hence the signal, and consequently its power, will be wasted.

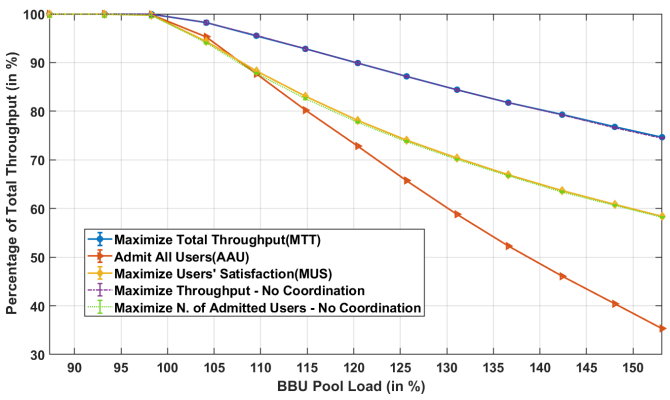
In the next subsections, we evaluate the performance of the three different scheduling solutions with respect to these four metrics. We limit the study and analysis to only one TTI instance, and we leave the multiple instants scenario for a future work. Also, we compare our approaches to two other basic approaches in the literature [4], that do not consider any coordination between radio and computing schedulers. Their objectives are to maximize throughput and to maximize 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. Total System Throughput

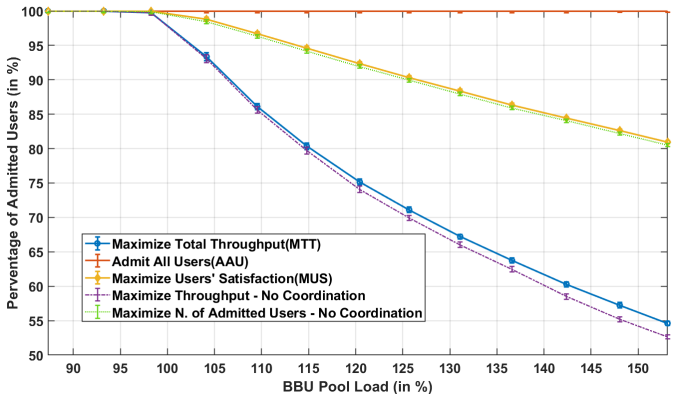
Fig. 2(a) shows the overall throughput obtained by each of the proposed approaches as a function of the BBU pool load. We note that this throughput is normalized with respect to the total throughput's demand that corresponds to the usage of the maximum MCS index by all users. The MTT solution clearly outperforms the other two coordination solutions in terms of total throughput when the BBU pool load surpasses 100%. In contrast, when the BBU pool load is less than 100%, the different solutions provide the same performance as the BBU pool is still able to 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 in terms of offered throughput 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 decreases the throughput) of many users so they may be scheduled on the BBU pool. Consequently, this severely degrades the total system throughput.

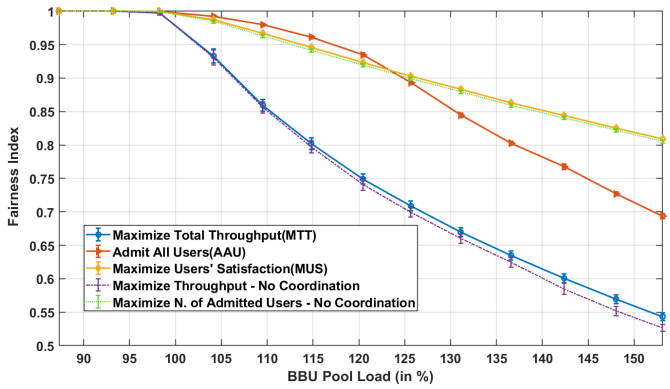
The MUS policy scores in-between results compared to the other polices. When comparing MTT and MUS policies to the non-coordination schemes, we find that MTT results almost coincide with Maximizing Throughput objective, and MUS results almost coincide with those found for Maximizing the Number of Admitted Users; the proposed coordination brings a very slight improvement less than 1%. In fact the coordination policies present an advantage over the non-coordination schemes since due to the allowed reassignment of MCS indexes, coordination schemes are able to 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



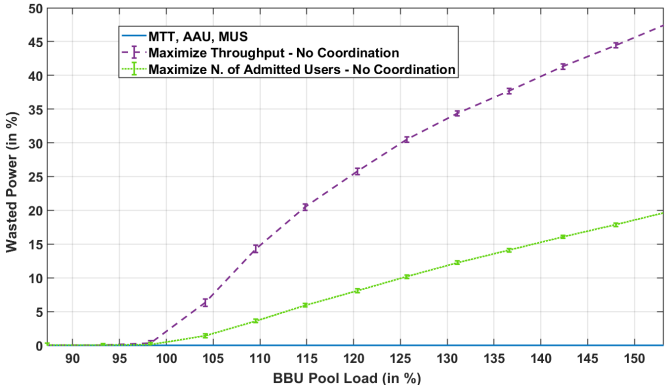
(a) Offered throughput (in %) as a function of BBU pool load



(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

Figure 2: Performance evaluation of the different scheduling solutions

making place for one or more additional users. As can be seen in constraint (4), this additional degree of liberty compared to the non-coordination schemes (where a single MCS may be assigned) can only be beneficial, producing larger or equal throughput and number of served users.

B. Number of Admitted Users

Next, Fig. 2(b) shows the percentage of admitted users as a function of the number of the BBU pool load; this percentage being relative to the total number of users from all RRHs. Intuitively, the AAU policy ensures the admission of all users (i.e., percentage of admitted users is always 100%) and it clearly outperforms all other solutions according to this performance metric. With respect to AAU, we see that the performance of MTT policy drops, admitting only 49.5% of users for an RRH number of 140 per BBU pool (ie. equivalent to a BBU pool load of 153%). Clearly, MTT prefers to admit less users with higher throughput than to admit more users with low throughput. For the third policy (MUS), its performance drops gradually until it reaches 85% when the BBU pool load is 153%. Again, the performance of the third policy comes in-between that of the other two policies. 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. 2(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 starts to decline. The MUS policy outperforms all the other policies in terms of overall fairness as it tries to maximize users' satisfaction rate. On the contrary, the MTT policy is the least fair among the coordination policies as it favors users who are able to 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; the addition of more users to the system worsens the fairness index since a lot of 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 from the system since it is impossible to process their data as they use the maximum MCS. 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 start to 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 not possible 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. 2(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 which will not be processed before the HARQ deadline due to lack of processing resources and that will, thus, be retransmitted. We consider that all users use a constant transmit power P_t for signal transmission. In the coordination schemes we present, only users whose frame can be processed (eventually with a reduced MCS index) before the HARQ deadline are going to send their 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 even knowing whether the BBU pool will be able to process users' data or not. 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 till it reaches 48% and 20% respectively when the BBU load is 153%. Here we notice a significant benefit of the proposed coordination between radio and CPU scheduling as it saves a significant amount of power.

E. MCS Selection Distribution

In order 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. 3 the cumulative distribution function of the selected MCS indexes when the number of RRHs per BBU pool is equal to 140, 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 other curves are concerned with only the users who are admitted. The results show that AAU policy, in order to ensure the admission of all the users in the BBU pool, forces the radio scheduler to strongly decrease the MCS indexes of users. In particular, the median value in AAU policy is 0, meaning that 50% of users operate with the lowest MCS index that is 0 and on the other hand there exists no user under this policy who operates with an MCS index higher than 6. Looking at the MTT policy, we notice that it favors the selection of 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, hence leaving those with low MCS indexed with no resources. On the other hand, the CDF of MUS policy is similar to that of Users' MAX MCS Distribution. This justifies its fairness, since the similarity in the curves indicates that MUS policy tries to assign to each user an MCS closer to its maximum allowed one; it attempts to make users fully satisfied as much as possible. With a probability greater than 0.6, MUS policy is going to select an MCS between 4 and 12.

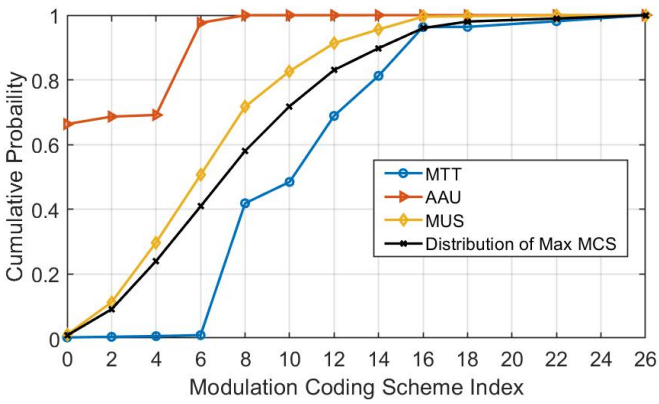


Figure 3: Cumulative Distribution Function of the users' MCS indexes when the RRHs number is equal to 140

As a 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 chooses in-between MCS indexes.

Moreover, we have shown that the proposed coordination scheme brings improvements especially in terms of reducing the amount of wasted power. However, a practical implementation of the proposed coordination cannot be based on solving an optimization problem that may require heavy computational resources. For that, it is necessary to propose low-complexity heuristics that are able to achieve a performance close to that of the ILP coordination solutions and to allocate resources in real-time. In the following section, we discuss 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. Given that solving an integer linear programming requires a lot of computational resources, and that it could be computationally infeasible to perform it in real-time, it might be necessary to switch our focus to low-complexity algorithms that utilize the coordination principle, and that are able to output sub-optimal allocation in a short time. For this reason, we propose and evaluate three heuristics that can be used as alternatives to the ILP-based algorithms proposed in section III-B.

A. The proposed heuristics

In this section, we propose three heuristics that consider the adjustment of the MCS indexes of users. We observed that all the heuristics decreased the run-time by more than 99% in comparison with their ILP counterparts. 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 margin variable $AdjMargin$ is initialized to zero; this variable sets a limit on 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

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 according 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 : $x^c_{r,u,m}$, $\forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C}$.

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.

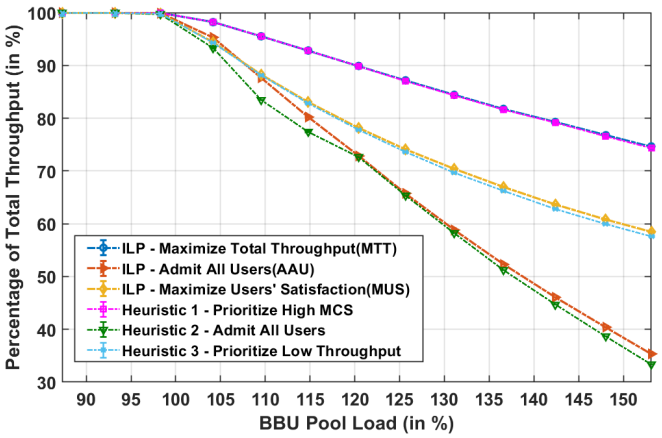
- *Heuristic 2 - Admit All Users*: Apply a two-level sorting to all users from all RRHs; firstly in ascending order of maximum MCS-Index; then in ascending order of maximum achievable throughput. a variable called $SysMCS$ is initialized to zero. This variable defines a limit on the MCS index that can be used by all users. Afterwards, 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

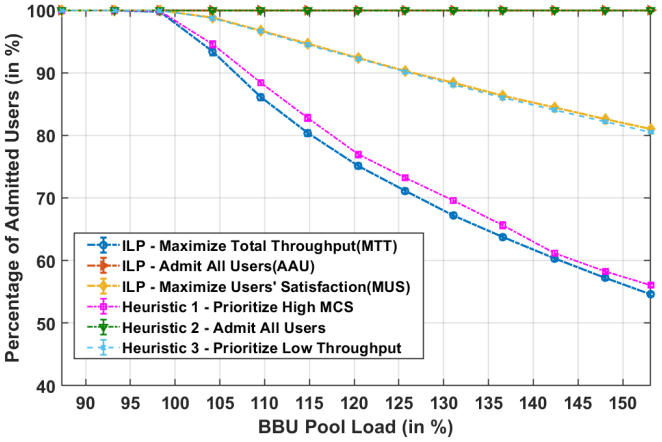
The proposed heuristics are evaluated and compared to the ILP-based policies of section III-B with respect to the same metrics of section IV: Total Throughput, Number of Admitted Users, and Fairness Index. The results are depicted in figures 4(a), 4(b), and 4(c), respectively. We clearly note from the figures that:

- *Heuristic 1 - Prioritize High MCS* shows very close results to those obtained by the first ILP problem, MTT, especially in terms of the throughput metric. However, it outperforms MTT with respect to the other two metrics: the number of admitted users and fairness metrics. In comparison with MTT, this heuristic can score up to 3% of improvement with respect to the percentage of admitted users metric and up to 0.026 of improvement with respect to 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 total throughput 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 5%.
- *Heuristic 3 - Prioritize Low Throughput* has more or less a similar performance in comparison to MUS policy with respect to all metrics.

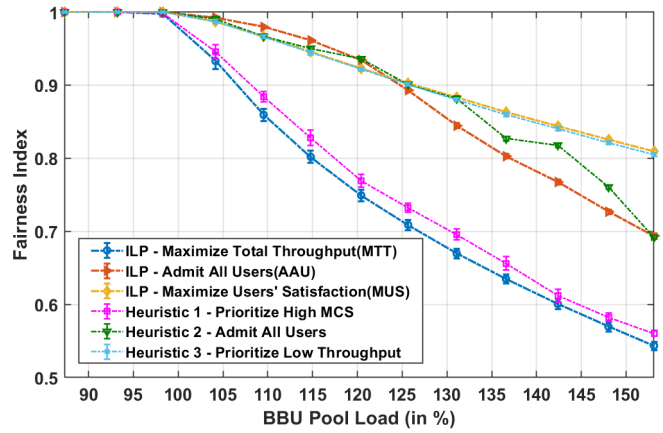
Each of the three heuristics, in comparison with its corresponding ILP counterpart, has managed to score close results with respect to all performance metrics. We recall that the three heuristics score 0% with respect to the metric of wasted power. Just like the ILP coordination solutions, users who are informed that the BBU can't process their data will not transmit, and thus will save the transmission power. In short, the proposed heuristics can serve as practical replacement for the high-complexity ILP algorithms, and can be implemented by the mobile operators to apply real-time scheduling.



(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

Figure 4: Comparison of the performance of the heuristics in comparison to ILP problems

VI. Conclusion

In this paper, we investigated three ILP policies that implement coordination between radio and computing schedulers in the context of Cloud-RAN. Motivated by the fact that the data processing time strongly depends on the MCS index selected for transmission, 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 at maximizing total throughput, admitting all users, and maximizing users' satisfaction, respectively. We have evaluated them according to different performance metrics. Results show that the proposed coordination achieves an important improvement by significantly reducing the amount of wasted transmission power and brings a slight but systematic improvement with respect to the other metrics. Among the ILP cooridnation policies, the MUS policy is the fairest; it achieves in-between values of throughput and number of allocated users in comparison with the other coordination policies. In addition, we propose three low-complexity heuristics and compare 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. As a future work, we aim to extend our study and evaluate the performance at the MAC layer taking into account the transmission errors and subframe re-transmission at the MAC-layer level.

Acknowledgment

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".

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] E. T. . . V12.3.0. (2014) LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures 3GPP TS 36.213 v.12.3.0 Rel.12.

[11] 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.

[12] 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.