Scheduling Algorithms for 5G Networks with Mid-haul Capacity Constraints

Abhishek Sinha*, Matthew Andrews†, Prasanth Ananth†
*Dept. of Electrical Engineering, IIT Madras, Chennai, India
†Nokia Bell Labs, Murray Hill, NJ, USA
Email: * abhishek.sinha@ee.iitm.ac.in, † matthew.andrews@nokia-bell-labs.com, † prasanth.ananth@nokia-bell-labs.com

***Abstract*—We consider a virtualized RAN architecture for 5G networks where the Remote Units are connected to a central unit via a mid-haul. To support high data rates, the mid-haul is realized with a Passive Optical Network (PON). In this architecture, the data are stored at the central unit until the scheduler decides to transmit it through the mid-haul to an appropriate remote unit, and then over the air at the same slot. We study an optimal scheduling problem that arises in this context. This problem has two key features. First, multiple cells must be scheduled simultaneously for efficient operation. Second, the interplay between the time-varying wireless interface rates and the fixed capacity PON needs to be handled efficiently. In this paper, we take a comprehensive look at this resource allocation problem by formulating it as a utility-maximization problem. Using combinatorial techniques, we derive useful structural properties of the optimal allocation and utilize these results to design polynomial-time approximation algorithms and a pseudo-polynomial-time optimal algorithm. Finally, we numerically compare the performance of the proposed algorithms to heuristics which are natural generalizations of the ubiquitous Proportional Fair algorithm.**

I. INTRODUCTION

Two inexorable trends will have a significant impact on the future of 5G wireless access. The first is the trend towards denser small cells with deeper fiber, which is sometimes described using the slogan “long wires and short wireless”. The second is the trend towards virtualized Radio Access Network (vRAN) architectures in which part of the processing and network intelligence (including scheduling) takes place in the central units (CUs) located in a cloud data center (sometimes called an *edge cloud*) and then the data are carried over a transport network called *mid-haul* to a set of remote units (RUs).

A passive optical network (PON) is ideally suited for such mid-haul due to its high capacity, lower cost, and ability to reuse the existing fiber-to-the-x (FTTx) distribution networks. However, if we utilize such an architecture, then we need to ensure that the scheduling decisions in the central units respect the limited PON capacity, in addition to the time-varying air interface data rate. The goal of this paper is to investigate how such scheduling can be carried out efficiently.

There are many variants of the vRAN architecture that differ based on how the processing is split between the CUs and the RUs. At a high-level, we can categorize these options into two types. In a *front-haul* architecture, all processing right down to the baseband takes place in the edge cloud. On the contrary, in this paper, we will be concerned with the so-called *mid-haul* architecture [1], [2], where some of the higher-layer processing takes place in the edge cloud while the lower physical layer processing takes place at the RUs. Hence, the mid-haul architecture requires less PON bandwidth compared to the front-haul architecture. However, the mid-haul bandwidth requirement changes with time depending on the actual amount of user traffic and the instantaneous wireless channel conditions. In this paper, we address the following question- How should the central units schedule the wireless transmissions at RUs efficiently in the *full-buffer* traffic regime so that the *fixed* PON capacity constraint and the *time-varying* wireless interface rate constraints are satisfied (see Figure 1)?

Intuitively, the centralized scheduler must take into account the limited PON capacity in addition to the instantaneous air interface channel conditions for efficient operation. To minimize latency (*e.g.,* in the case of URLLC traffic), the scheduling should be done in such a way that there is *no queue build-up* at the RUs.

The conflicting nature of the constraints makes this resource allocation problem challenging to solve. For the wireless air interface, the fundamental resource units are the *resource blocks* (RBs), which give rise to time-varying bit rates according to the dynamic wireless channel conditions. On the other hand, for the PON, the fundamental resource units are *fixed-capacity* PON slices. As a concrete example, the air interface scheduler may wish to serve a user that is in a good channel condition. However, it may not be able to do that if the PON cannot handle the resulting data rate. In this paper, we undertake a comprehensive study of this problem and develop algorithms with provable guarantees to solve it efficiently.
The rest of the paper is organized as follows:

- In Section II, we formulate the problem by using the theory of gradient ascent over time-varying channels [3]. This methodology decomposes the long-run average utility objective into slot-by-slot local objectives.
- In Section III, we give an illustrative example to show why the standard greedy algorithms (including, *e.g.*, the Proportional Fair scheduling) fail to provide an optimal solution. The fundamental difficulty is that greedy approaches cannot optimally handle the mismatch between the air interface constraint and the PON capacity constraint.

* Part of this work was completed when the first author was interning at Nokia Bell Labs, Murray Hill, NJ, USA. The work of the first author is partially supported by a grant from Qualcomm.

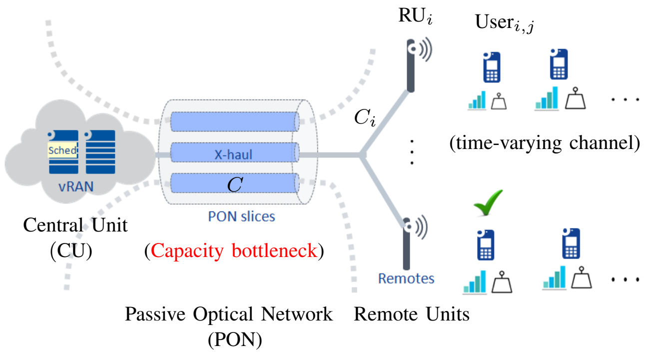

Fig. 1: Schematic of a split-processing vRAN architecture

- In Section IV, we present an efficient algorithm that gives the optimal wireless rate allocations for a fixed RB assignment to the user. This algorithm is used in our later developments.
- In Section V, we present two algorithms for the special case when only the overall PON capacity constraint is binding. In particular, we present a polynomial-time 2-approximation algorithm based on LP-rounding. This algorithm exploits the special structure of the basic feasible solutions of the associated LP. We also provide an optimal Dynamic Programming (DP) algorithm that runs in pseudo-polynomial time.
- In Section VI, we present a greedy 2-approximation matroid-based algorithm for the general case, where the individual RU-specific capacity constraints, as well as the overall PON capacity constraint are active.
- In Section VII, we present two natural heuristics which are inspired from the ubiquitous Proportional Fair algorithm.
- In Section VIII, we evaluate the proposed algorithms via numerical simulation and examine how they compare to the heuristics.
- Finally, Section IX concludes the paper after a brief discussion on related works.

II. Problem Formulation

System Model: We consider a split-processing vRAN architecture as shown in Figure 1. There are m Remote Units (RUs) that transmit data over the air to the wireless end users connected to it. A Passive Optical Network (PON) connects a Central Unit (CU) to the Remote Units (RUs). At each time slot, each RU can transmit data on κ Resource Blocks (RBs), each of which corresponds to a set of contiguous OFDM carriers. At every RU, each RB can be assigned to at most one user at a slot. We assume that inter-RU interference is negligible, which is a reasonable assumption given the advent of the CoMP technology [4].

Traffic Model: For simplicity, we focus on the downlink traffic only. We assume that the users are infinitely backlogged, i.e., the users' respective data buffers in the CU are always full. In order to keep the latency small, data is *not buffered* in the Remote Units. Hence, all scheduled data from the CU must be delivered to the users through the PON and the wireless interface at the RUs at the *same* slot. All scheduling decisions are assumed to be made by the CU.

Decision Variables and Constraints: We use the symbol i to index the RUs, the pair (i,j) to index the j^{th} user associated with the i^{th} RU (denoted by RU$_i$), and k to index the RBs. Let C be the total capacity of the PON mid-haul, and let C_i be the capacity of the optical fibers connecting RU$_i$ to the PON (see Fig. 1). Let $\gamma_{ijk}(t)$ denote the instantaneous air-interface rate for the user (i,j) on the RB k at slot t. Let $x_{ijk}(t) \in \{1,0\}$ be a binary *decision variable* representing whether or not the RB k at RU$_i$ is assigned to the user (i,j) at slot t. Let $y_{ijk}(t)$ be a non-negative *decision variable* denoting the rate allocated to the user (i,j) on the RB k at slot t.

The limited PON capacity and air-interface rates enforce the following constraints on the instantaneous decision variables $\boldsymbol{x}(t), \boldsymbol{y}(t)$:

$$y_{ijk}(t) \leq \gamma_{ijk}(t)x_{ijk}(t), \quad \forall i,j,k, \tag{1}$$

$$\sum_j x_{ijk}(t) \leq 1, \quad \forall i,k, \tag{2}$$

$$\sum_{jk} y_{ijk}(t) \leq C_i, \quad \forall i, \tag{3}$$

$$\sum_{ijk} y_{ijk}(t) \leq C, \tag{4}$$

$$x_{ijk}(t) \in \{0,1\}, \; y_{ijk} \geq 0, \quad \forall i,j,k. \tag{5}$$

Discussions on the constraints (1)-(5)*:* The inequality (1) reflects the fact that the allocated rate $y_{ijk}(t)$ can be non-zero only if the RB k is assigned to the user (i,j) at time t. Moreover, due to time-varying nature of the air-interface rate, the allocated data rate at slot t can be at most $\gamma_{ijk}(t)$ for that assignment. Inequality (2) states that at most one user may be assigned to any RB k on RU$_i$. Inequalities (3) and (4) denote the mid-haul capacity constraints. The constraint (5) simply denotes the fact that the variables $x_{i,j,k}(t)$ are binary and the allocated rates are non-negative.

Special Case: In general, the capacity of optical fibers decrease sharply with their length [5]. Since the distance between the PON remote end-point and any RU is much shorter than the size of the PON itself, it is often the case that $C_i >> C, \forall i$. To exploit this fact in designing algorithms, we pay particular attention to the important special case where the RU specific constraints (3) are relaxed (effectively by setting $C_i = \infty, \forall i$), and the system is limited by the overall PON capacity constraint (4) only (see Section V).

Objective: In this paper, we formulate the resource allocation problem as a utility maximization problem. Let $\bar{r}_{ij}$ be the long-term average data rate for the user (i,j), i.e.,

$$\bar{r}_{ij} = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_k y_{ijk}(t).$$

Let $U(\cdot)$ be a strictly concave smooth utility function. Our objective is to find a scheduling policy that maximizes the sum-utility of all users defined as follows:

$$U(\bar{\boldsymbol{r}}) = \sum_{ij} U(\bar{r}_{ij}). \tag{6}$$

Following the development in [3], for each user (i, j), we first define an exponentially smoothed long-term service rate $R_{ij}(t)$, which evolves as follows:

$$R_{ij}(t+1) = (1-\beta)R_{ij}(t) + \beta \sum_k y_{ijk}(t), \quad R_{ij}(1) = 0, \quad (7)$$

where $\beta > 0$ is a small positive constant. From [3], it follows that the long-term objective (6) is maximized by finding the instantaneous decision variables $\boldsymbol{x}(t), \boldsymbol{y}(t)$ at each slot t that solves the following problem, referred to as SINGLE SHOT:

$$\max_{\boldsymbol{x}(t),\boldsymbol{y}(t)} \sum_{ij} U'(R_{ij}(t)) \sum_k y_{ijk}(t), \quad (8)$$

subject to the constraints (1)-(5). We therefore focus on the SINGLE-SHOT problem for the slot t for the remainder of the paper.

For the common case of logarithmic utility, in which $U(x) = \log x$, the above per-slot problem (8) becomes:

$$\max_{\boldsymbol{x}(t),\boldsymbol{y}(t)} \sum_{ij} \frac{\sum_k y_{ijk}(t)}{R_{ij}(t)}, \quad (9)$$

subject to the constraints (1)-(5). For concreteness, we will use the objective (9) throughout the paper, but all our results apply to any strictly concave smooth utility function $U(\cdot)$.

It should be noted that in the case of inelastic traffic, where the objective is to ensure network stability while achieving throughput-optimality, a standard algorithm is MAX-WEIGHT [6]. In the MAX-WEIGHT algorithm, one is required to solve an identical problem to SINGLE SHOT at every slot, where the factor $R_{ij}(t)^{-1}$ in the objective (9) is replaced with the corresponding queue-length. Hence, the algorithmic techniques that we develop for SINGLE SHOT directly apply to the case of MAX-WEIGHT algorithm for inelastic traffic.
Since, we solve the SINGLE SHOT problem at every slot t, to avoid notational clutter, we will be dropping the time-argument t from all variables henceforth.

III. SUB-OPTIMALITY OF THE PROPORTIONAL FAIR SCHEDULER

For solving the SINGLE SHOT problem (9), the well-known Proportional Fair scheduler (PF) assigns the RB k at RU_i to the user (i, j) that maximizes the index γ_{ijk}/R_{ij} [7]. It does so without taking into account the capacity constraints (3)-(4) for the PON. We prove that the PF algorithm is not optimal due to the presence of the capacity constraints.

Lemma 1 (Sub-Optimality of PF): The Proportional Fair (PF) scheduler is not optimal for the SINGLE SHOT problem (9), in general.

Proof: Our counter-example has one RU with two users and four RBs. Let $C = 7$. Note that, we don't need to specify the separate C_i values since we have only one RU. Let the aggregate rates for the two users at slot t be,

$$R_{00} = 1 \quad R_{01} = 2.$$

Let the instantaneous channel rates be,

$$\gamma_{00k} = 1 \quad \gamma_{01k} = 4 \quad \forall k.$$

Since $\gamma_{01k}/R_{01} = 2 > 1 = \gamma_{00k}/R_{00}$, PF will pick user 1 for every RB, i.e. $x_{01k} = 1$ and $x_{00k} = 0$ for all k. Given the PON capacity constraint, we choose the y values such that $\sum_k y_{01k} = 7$ and $\sum_k y_{00k} = 0$. Hence, the total objective value for SINGLE SHOT is $7/2$.

A better solution would put user 0 on 3 RBs and user 1 on a single RB. In this case we have $x_{00k} = 1$ and $y_{00k} = 1$ for $k < 3$ and $x_{013} = 1$ and $y_{013} = 4$. The total objective for this solution is $\frac{3}{1} + \frac{4}{2} = 5$. ■

IV. STRUCTURAL RESULTS

In some sense, the difficulty of maximizing (9) stems from the fact that the optimal solution may split the total RB allocation between the users with high values of $1/R_{ij}$ and the users with high values of γ_{ijk}/R_{ij}. Recall from Lemma 1 that, if the PF algorithm violates the capacity constraints, then it might lead to a suboptimal allocation. We now state an intuitively obvious result that if this violation does not happen then, PF is, in fact, optimal.

Lemma 2: Suppose that with the PF allocation, we can set $y_{ijk} = \gamma_{ijk}x_{ijk}, \forall i, j, k$, without violating the capacity constraints (3)-(4). Then PF achieves optimality.

Proof: Follows from the observation that the maximum objective value that we can obtain from RB k at RU_i is $\max_j \gamma_{ijk}/R_{ij}$. If the PF Algorithm achieves this value without violating the capacity constraints, then it is optimal. ■

In order to proceed further, Lemma 3 presents a simple and efficient strategy that allows us to determine the optimal allocation $\boldsymbol{y}_{\text{opt}}$ for any given feasible assignment $\boldsymbol{x}$. As a consequence of Lemma 3, the problem (9) reduces to determining the optimal assignment $\boldsymbol{x}^*$, *i.e.*, the identity of the user j to which the RB k on RU_i should be assigned. The optimal amount of service y_{ijk} that the user (i, j) then receives is determined by this lemma.

Lemma 3: Suppose that we are given a set of binary $\boldsymbol{x}$ values that satisfy the feasibility constraint (2). We can find the optimal rate allocation $\boldsymbol{y}_{\text{opt}}$ for this set of $\boldsymbol{x}$ via the following iterative scheme:

Algorithm 1 Optimal Rate Allocation ($\boldsymbol{y}_{\text{opt}}$) for a given RB Assignment Profile ($\boldsymbol{x}$)

1: Set $\boldsymbol{y} \leftarrow \mathbf{0}$.
2: Order the (ijk) triples in decreasing order of the value of $1/R_{ij}$.
3: Go through each triple sequentially in order. When considering the triple (ijk), set

$$y_{ijk} \leftarrow \min\{\gamma_{ijk}x_{ijk}, C - \sum_{i'j'k'} y_{i'j'k'}, C_i - \sum_{j'k'} y_{ij'k'}\}.$$

4: Set $\boldsymbol{y}_{\text{opt}} \leftarrow \boldsymbol{y}$.

Proof: See Appendix A. ■

V. Algorithms for Single Shot with an overall PON Capacity Constraint

In this Section, we consider the special case where the separate RU-specific capacity constraints (3) are relaxed (by effectively setting $C_i = \infty, \forall i$). Thus the system is capacity-limited by the PON constraint (4) only. As discussed in Section II, this is a practically relevant case when the PON size is much larger than the RU-to-PON access distances.
It is not hard to see that, this special case is algorithmically equivalent to the scenario where there is only one RU, and all UEs are associated with this RU. Thus, to avoid notational clutter, we drop the RU index i throughout this Section. We start with the following definition:

Definition 1: A feasible rate allocation vector $\boldsymbol{y}$ is called Discrete if $y_{jk} = x_{jk}\gamma_{jk}, \forall k$.

In other words, in a Discrete allocation either the RB is allocated the maximum wireless rate given by the wireless interface rate (γ_{jk}) or it is not allocated any rate at all.

Definition 2: A feasible rate allocation vector $\boldsymbol{y}$ is called Almost Discrete if $y_{jk} = x_{jk}\gamma_{jk}$ for all RBs, possibly excepting *at most one* RB.

The significance of the above definition is borne out by the following Theorem.

> *Theorem 4:* There exists an optimal solution to Single Shot which is Almost Discrete.

Proof: Follows directly from Lemma 3 since it implies that once the $\boldsymbol{x}$ values are set, we can find the optimal $\boldsymbol{y}$ values by simply going through each of them in decreasing the order of $1/R_j$. Excepting the last one, each one is filled up to an amount γ_{jk} before moving on to the next one. ■

Note: It is also possible to prove Theorem 4 without appealing to Lemma 3. See Appendix B for an alternative proof using combinatorial properties of the associated LP.

In the following, we exploit the result in Theorem 4 to design a polynomial-time approximation algorithm to the Single Shot problem with an overall PON capacity constraint.

A. Poly-time 2-Approximation Algorithm Rounding-AD

We now design an LP-based algorithm Rounding-AD (AD stands for Almost Discrete) for approximately solving the Single Shot problem. By substituting $y_{jk} \leftarrow \gamma_{jk}x_{jk}$ and relaxing x_{jk} to take any real number in the interval $[0, 1]$, we obtain the following LP relaxation to (9):

$$\max \sum_{jk} x_{jk}\frac{\gamma_{jk}}{R_j} \tag{10}$$

subject to,

$$\sum_{j} x_{jk} \leq 1, \quad \forall k, \tag{11}$$

$$\sum_{jk} \gamma_{jk}x_{jk} \leq C, \tag{12}$$

$$\boldsymbol{x} \geq 0. \tag{13}$$

Call the previous relaxed Linear Program RLP. The following Theorem shows that an optimal solution to RLP is "close" to being an all-integral solution.

> *Theorem 5:* An optimal solution to RLP allocates every RB to at most one user, excepting, possibly *at most* one RB (which is shared between two users).

Proof: By introducing the non-negative auxiliary variables $\zeta_k, \forall k$ in (11) and ξ in (12), we obtain the following set of equivalent constraints:

$$\sum_{j} x_{jk} + \zeta_k = 1, \quad \forall k, \tag{14}$$

$$\sum_{jk} \gamma_{jk}x_{jk} + \xi = C, \tag{15}$$

where $\boldsymbol{x}, \boldsymbol{\zeta}, \xi \geq 0$. Next, recall that. for any LP, an optimal solution (also known as a **B**asic **F**easible **S**olution (**BFS**)), is always obtained at some vertex of the polytope defined by the constraints [8]. Let the total number of RBs be κ. Since there are a total of $(\kappa + 1)$ equality constraints taken together in the equality constraints (14)-(15), at most $(\kappa + 1)$ variables could be strictly positive in any **BFS**. Also, since the RHS of the constraints in (14) are positive, it follows that there is *at least one* strictly positive variable per equality constraints (14). This implies, by the pigeonhole principle, that, in an optimal solution to RLP, there could be at most one RB k_1 which has been allocated to two users $j_1, j_2, (j_1 \neq j_2)$ (i.e., $x_{j_1k_1}x_{j_2k_1} > 0$). Moreover, all other RBs have been allocated to at most one user in the optimal solution. ■

Note that, if the optimal solution to RLP contains no fractional variable, then it indeed yields an **optimal solution** to Single Shot. Finally, we use Theorem (5) to construct a 2-approximation algorithm for Single Shot.

□ *LP-based Polynomial-time 2-Approximation algorithm:* The (possible) non-integral optimal solution to RLP may be converted to a feasible 2-approximate optimal solution to Single Shot. Let the value of the optimal solution to RLP and the original Single Shot problem be denoted by OPT′ and OPT respectively. It is obvious that any feasible solution to Single Shot may be used to easily construct a feasible solution to RLP with the same objective value. Hence, we readily have

$$\mathsf{OPT}' \geq \mathsf{OPT}. \tag{16}$$

Next, let the contribution to the total objective value OPT′ in Eqn. (10) by the standalone RBs ($x_{jk} = 1$ for some j), and the (possible) one shared RB be I, F' respectively, where $\mathsf{OPT}' = I + F'$. The maximum contribution to the objective (9) in the Single Shot problem that we can obtain from any single RB, considering it separately, is $F_{\max} = \max_{j,k} \frac{1}{R_j} \min\{\gamma_{jk}, C\}$. Clearly, $F_{\max} \geq F'$. Finally, we choose the solution corresponding to the *maximum* of I and

$F_{\max}$. It is clearly a feasible solution to SINGLE SHOT and

$$\begin{aligned} \max\{I, F_{\max}\} &\geq \max\{I, F'\} \\ &\geq \frac{1}{2}(I + F') \\ &= \frac{1}{2}\mathsf{OPT}' \geq \frac{1}{2}\mathsf{OPT}. \end{aligned} \quad (17)$$

Eqn. (17) shows that the above LP-based scheme is a 2-approximate poly-time algorithm to the SINGLE SHOT problem with an overall PON capacity constraint. We summarize the above algorithm in Algorithm 2 below:

Algorithm 2 LP-based 2-Approximation Algorithm for SINGLE SHOT

1: Find the maximum possible objective value (9) obtainable by using a *single* RB, i.e.,

$$F_{\max} = \max_{j,k} \frac{1}{R_j} \min\{\gamma_{jk}, C\}. \quad (18)$$

2: Solve the Linear Program RLP (10). Let I be the objective value obtained by the standalone RBs (*i.e.,* for which $x_{jk} = 1$ for some j) in its optimal solution.

3: Choose the solution corresponding to the maximum of I and $F_{\max}$.

Although the algorithm ROUNDING AD has been shown to be a 2-approximation algorithm, our numerical simulations in Section VIII reveal that ROUNDING AD is likely to perform near-optimally in practice. In the following, we design a Dynamic Programming-based *Optimal* algorithm to SINGLE SHOT. However, unlike the previous LP-based 2-approximation algorithm, the Dynamic Program runs in pseudo-polynomial-time, and hence, it is less efficient than ROUNDING AD.

B. A DP-based Pseudo-Polynomial-time Optimal Algorithm

Without any loss of generality, we may assume that the capacity C and the wireless interface rates $\{\gamma_{jk}\}$ are integers. Then, Theorem (4) readily implies that the optimal allocated rates y_{jk}'s are also integers for all j, k. Hence, without any loss of optimality, we may consider the following discrete range $\mathcal{R}_{jk}$ for the decision variable y_{jk}:

$$\mathcal{R}_{jk} = \{0, 1, 2, \ldots, \gamma_{jk}\}, \quad \forall j, k. \quad (19)$$

Let κ denote the total number of RBs. Arrange the RBs in some order $(\text{RB}_1, \text{RB}_1, \ldots, \text{RB}_\kappa)$, and consider them adding to the solution one-by-one in this sequence. Let $V(M, k)$ denote the maximum objective value (9) obtained by using only the first k RBs with a PON of total capacity M. Then, as explained below, we have the following Dynamic Programming recursion for $V(M, k)$:

$$V(M,k) = \max_j \max_{y_{jk} \in \mathcal{R}_{jk}, y_{jk} \leq M} \big(\frac{1}{R_j} y_{jk} + V(M - y_{jk}, k-1)\big), \quad (20)$$

with $V(M, 0) = 0, \forall M$.

Optimality: The above recursion (20) may be obtained as follows. Consider the k^{th} RB and suppose that it is assigned to the j^{th} user and allocated a rate of y_{jk}. Hence, RB_k contributes a value of $\frac{y_{jk}}{R_j}$ towards the objective (9). With this assignment, we are left with the first $k-1$ RBs with a usable PON capacity of value $M - y_{jk}$, which, by the definition of $V(\cdot, \cdot)$, contributes a total value of $V(M - y_{jk}, k-1)$ to the objective (9) in an optimal allocation. Hence, $V(M, k)$ is found by optimizing the total objective $\big(\frac{1}{R_j} y_{jk} + V(M - y_{jk}, k-1)\big)$ over the choice of user assignment j and feasible rate allocation $y_{jk} \in \mathcal{R}_{jk}$ for the k^{th} RB. Moreover, since the total PON capacity is M, the variable y_{jk} can only assume those values the feasible set $\mathcal{R}_{jk}$ which are at most M. This proves optimality of the DP recursion (20). We summarize the DP algorithm below in Algorithm 3.

Algorithm 3 Optimal Dynamic Program for SINGLE SHOT

Set $V(M, 0) \leftarrow 0, \quad \forall M = 0, 1, 2 \ldots, C.$

for $k = 1, 2, \ldots, \kappa$ **do**

  for $M = 0, 1, \ldots, C$ **do**

$$V(M,k) = \max_j \max_{y_{jk} \in \mathcal{R}_{jk}, y_{jk} \leq M} \big(\frac{1}{R_j} y_{jk} + V(M - y_{jk}, k-1)\big).$$

  end for

end for

Return $V(C, \kappa)$.

VI. APPROXIMATION ALGORITHM FOR THE GENERAL **Single Shot** PROBLEM

In this Section, we present an approximation algorithm for the general SINGLE SHOT problem, in which both the overall PON capacity constraint (4), as well as the individual RU-specific capacity constraints (3) are active. Our proposed algorithm is a greedy 2-approximation algorithm called MATROID which is based on the theory of optimizing a sub-modular function over a partition matroid [9].

Algorithm MATROID

For a feasible binary assignment vector $\boldsymbol{x} = (x_{ijk})$ with $\sum_j x_{ijk} \leq 1, \forall i, k$, define a corresponding set S of RU-User-RB triples as follows:

$$S = \{(i, j, k), \quad \text{if} \quad x_{ijk} = 1\}.$$

Define the ground set $E = \{(i, j, k), \forall i, j, k\}$, and let $\mathcal{I}$ be the collection of all corresponding sets S for all feasible binary assignment vectors $\boldsymbol{x}$. We make the following claim:

Lemma 6: The system $(E, \mathcal{I})$ is a **partition matroid**.

Moreover, for a given RB-to-User assignment $S \in \mathcal{I}$, we can efficiently compute the optimal rate assignments y_{ijk} values by using Algorithm 1. Let $f(S)$ be the associated objective for the assignment S. We have the following lemma:

Lemma 7: The set function $f(\cdot)$ is submodular.

Proof: See Appendix D. ■

The greedy algorithm works by initializing S to a null set and then repeatedly choosing a feasible augmentation $\bar{S}$ of S

that maintains the feasibility constraint $\sum_j x_{ijk} \leq 1$ and which maximizes the increase in $f(S)$. The algorithm is summarized in Algorithm 4.

Algorithm 4 Greedy Algorithm for SINGLE SHOT

1: $S \leftarrow \phi$
2: **while** 1 **do**
3:   Find a feasible augmentation $\bar{S} \in \mathcal{I}$ of S that maximizes $f(\bar{S})$ subject to the constraint $|\bar{S} \setminus S| = 1$.
4:   **if** $f(\bar{S}) = f(S)$ **then**
5:     **break**
6:   **else**
7:     $S \leftarrow \bar{S}$
8:   **end if**
9: **end while**

Lemma 8: Algorithm 4 is a 2-approximation algorithm for SINGLE SHOT.

Proof: This is a direct result of the Fisher-Nemhauser-Wolsey [9] algorithm for maximizing a submodular function over a matroid. ■

Discussion: Although Algorithm 4 also applies to the case when there is an overall PON capacity constraint (Section V), the LP-based Algorithm 2 ROUNDING AD runs much faster than the Matroid-based Greedy Algorithm 4. This is because, each candidate augmentation in the step 3 of Algorithm 4 requires an invocation of Algorithm 1 for the evaluation of $f(\bar{S})$, which is costly.

VII. HEURISTIC ALGORITHMS

In this Section, we present two natural heuristics that provide a baseline for numerical comparison with our proposed optimal and approximation algorithms in the following Section.

• MAX-YIELD: This is the simplest adaptation of the traditional PF algorithm so that it respects the capacity constraints. The algorithm works by going through the RUs and RBs in decreasing order of the index $\max_j \gamma_{ijk}/R_{ij}$ sequentially and always picking the user j that maximizes the index γ_{ijk}/R_{ij}. At each step, the used and remaining capacity on the PON is tracked, and the algorithm stops when the available capacity is exhausted.

• MAX-VALUE: This algorithm works by going through the RUs and RBs in decreasing order of the index $\max_j \gamma_{ijk}/R_{ij}$ sequentially and always picking the user that maximizes $1/R_{ij}$. At each step, the used and remaining capacity on the PON are tracked, and the algorithm stops when the available capacity is exhausted. Note that MAX-YIELD tries to optimize the objective with respect to the wireless resources, and algorithm MAX-VALUE seeks to maximize the objective with respect to the PON capacity constraints.

VIII. SIMULATION RESULTS

Set-up: We numerically simulate the performance of the proposed scheduling algorithms over a service area of 1 sq. km serving 1000 wireless users via 100 RUs with overall one PON capacity constraint. We experiment with two different types of mid-hauls - one with a PON transport capacity of $C = 1$ Gbps and another with a PON capacity of $C = 1000$ Gbps. In the former case, the PON capacity is highly constraining, whereas in the latter case the PON capacity is hardly constraining. The users and the RUs are assumed to be distributed over the service area according to a two-dimensional Poisson Point Process. The wireless channel has a bandwidth of 20 MHz, and all RUs have omnidirectional antennas with transmit power of 24 dBm. The path-loss coefficient is $\alpha_{\text{LOS}} = 2.09$ for a line-of-sight transmit/receiver pair and $\alpha_{\text{NLOS}} = 3.75$ for a non-line-of-sight pair. The probability for a transmit/receiver pair to be line-of-sight is $p_{\text{LOS}} = 0.12$, if their separation is less than 200m and is zero otherwise. Each wireless channel has a been simulated with a random fading process using Jakes' model [10] with a maximum doppler shift of 10 Hz. The wireless fading, in turn, determines the air-interface rate-vector $\boldsymbol{\gamma}(t)$.

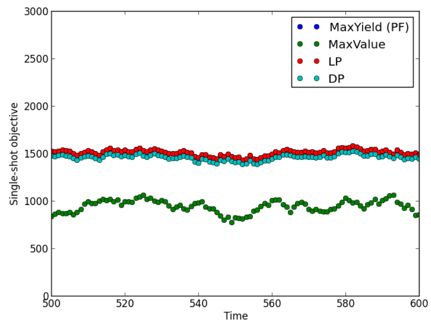

Fig. 2: SINGLE SHOT objective ($C = 1000$ Gbps)

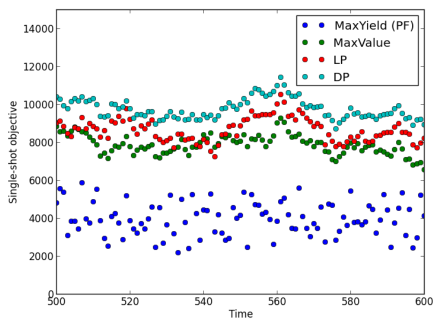

Fig. 3: SINGLE SHOT objective ($C = 1$ Gbps)

Results and Discussion: In Figures 2 and 3, we show the SINGLE SHOT objective achieved by MAX-YIELD, MAX-VALUE, the Dynamic Programming algorithm DP, and the LP-based 2-approximation algorithm ROUNDING-AD. We plot these values over 100 time slots (after a warm-up period). To fairly compare the efficacy of the above algorithms one a slot-by-slot basis. In our simulations, we assume that all algorithms use the same set of $R_{ij}(t)$ values (that are commonly driven by the MAX-YIELD algorithm).

For the case $C = 1000$ Gbps, the PON capacity is not constraining and hence, the MAX-YIELD algorithm is optimal.

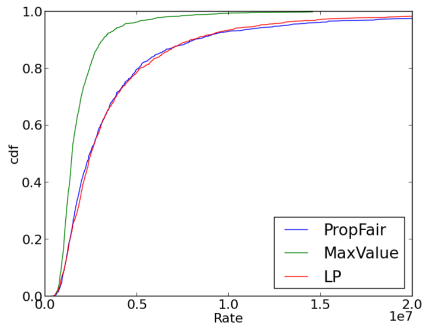

Fig. 4: Long-term user rate distribution ($C = 1000$ Gbps)

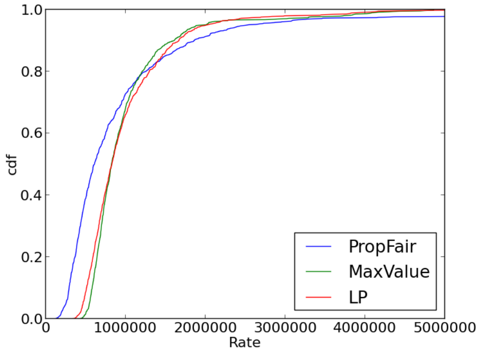

Fig. 5: Long-term user rate distribution ($C = 1$ Gbps)

In this case, we observe from Figure 2 that the DP and LP-based approaches are essentially optimal as well. (The points for LP coincide with (and therefore hide) the points for MAX-YIELD.) On the other hand, Figure 3 shows that for the case of $C = 1$ Gbps, a pure Proportional Fair approach would violate the PON capacity constraint, and hence, MAX-YIELD is not optimal. Moreover, MAX-VALUE is also not optimal either since it would not necessarily fill up the PON capacity. We see that both the DP and LP-based algorithms perform better than the heuristics. The fact that DP and LP-based algorithms work well in both cases illustrates the benefits of scheduling with an awareness of both the channel conditions and the PON capacity.

In Figures 4 and 5 we observe the overall user rate distribution when we run Proportional Fair (MAX-YIELD), MAX-VALUE and the LP-based 2-approximation algorithm ROUNDING AD. If $C = 1000$ Gbps, then ROUNDING AD has a similar rate distribution to the optimal Proportional Fair algorithm. If $C = 1$ Gbps, then ROUNDING AD outperforms Proportional Fair (MAX-YIELD) and MAX-VALUE algorithms for all except the users with the highest channel rates. This, in turn, means that ROUNDING AD leads to a higher value of the logarithmic utility function, which rewards fairness.

IX. Discussion and Related Work

In this paper, we have analyzed a scheduling problem that arises in the context of virtualized RAN architecture with a fixed capacity PON mid-haul. The importance of this type of scheduling problem is on the rise given the shift towards more flexible split-processing architecture for 5G wireless networks. We view our work as a natural extension of the large body literature on scheduling over time-varying channels [11], [12], [13], [14], [15], [16], [17], [18], [6]. This body of research introduced and influenced the popular Proportional Fair (PF) algorithm [19], [20], which is implemented in almost all of today's cellular network. Our algorithms lead to a different and more efficient scheduling than the usual PF algorithm when the PON mid-haul capacity is the bottleneck.

References

[1] T. Pfeiffer, "Next generation mobile fronthaul and midhaul architectures," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B38–B45, 2015.

[2] U. Dötsch, M. Doll, H. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 105–128, 2013.

[3] A. L. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation," *Operations research*, vol. 53, no. 1, pp. 12–25, 2005.

[4] L. Liu, J. C. Zhang, Y. Yi, H. Li, and J. Zhang, "Combating interference: Mu-mimo comp and hetnet," *J. Commun.*, vol. 7, no. 9, pp. 646–655, 2012.

[5] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical networks: a practical perspective*. Morgan Kaufmann, 2009.

[6] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE transactions on automatic control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[7] H. Kim and Y. Han, "A proportional fair scheduling for multicarrier transmission systems," *IEEE Communications Letters*, vol. 9, no. 3, pp. 210–212, March 2005.

[8] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[9] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—ii," in *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.

[10] J. WC Jr, "Microwave mobile communications," 1974.

[11] R. Agrawal and V. Subramanian, "Optimality of certain channel aware scheduling policies," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 40, no. 3. The University; 1998, 2002, pp. 1533–1542.

[12] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2415–2424.

[13] S. Borst and P. Whiting, "Dynamic rate control algorithms for hdr throughput optimization," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2001, pp. 976–985.

[14] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1. IEEE, 2003, pp. 321–331.

[15] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2064, 2001.

[16] X. Liu, E. K. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer networks*, vol. 41, no. 4, pp. 451–474, 2003.

[17] S. Shakkottai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in hdr," in *Teletraffic Science and Engineering*. Elsevier, 2001, vol. 4, pp. 793–804.

[18] ——, “Scheduling for multiple flows sharing a time-varying channel: The exponential rule,” *Translations of the American Mathematical Society-Series 2*, vol. 207, pp. 185–202, 2002.

[19] P. Viswanath, D. N. C. Tse, and R. Laroia, “Opportunistic beamforming using dumb antennas,” *IEEE transactions on information theory*, vol. 48, no. 6, pp. 1277–1294, 2002.

[20] A. Jalali, R. Padovani, and R. Pankaj, “Data throughput of cdma-hdr a high efficiency-high data rate personal communication wireless system,” in *Vehicular technology conference proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 3. IEEE, 2000, pp. 1854–1858.

[21] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

APPENDIX

A. Proof of Lemma 3

Proof: Let $\boldsymbol{y}^*$ denote the optimal solution for the assignment $\boldsymbol{x}$. Consider the (ijk) triples in the order above and consider the first triple (ijk) for which the $y_{ijk}(t)$ value according to the above algorithm is different from $y^*_{ijk}(t)$. Since the $y_{ijk}(t)$ have been made as large as possible subject to all of the constraints, it must be the case that $y^*_{ijk}(t) < y_{ijk}(t)$. We now increase in a continuous manner until $y^*_{ijk}(t) = y_{ijk}(t)$. In order to do this, we might have to decrease some other y^* values. It is not necessary to do this for $i'j'k'$ triples that have already been considered since $y_{ijk}(t)$ does not violate any constraints. If there is a value $y^*_{ij'k'}(t) > 0$ for a later triple $(ij'k')$ at the same RU then we decrease it until either it hits zero or all the constraints are satisfied. If there is no such $y^*_{ij'k'}(t)$ at the same RU then we do the same but for a later $y^*_{i'j'k'}$ value at a different RU. We can always find such a value since otherwise $y^*_{i'j'k'}(t) = 0$ for all later triples. This cannot be true if the y^* values satisfy the constraints since the y values represent a feasible solution.

Since we are decreasing $y^*_{i'j'k'}(t)$ for a later triple it must be the case that $1/R_{ijk}(t) \geq 1/R_{i'j'k'}$. Hence the objective function for the y^* values cannot get any worse as we make the changes. If we keep repeating the procedure then eventually the y^* values will equal the y values. This implies that the y values found from the above procedure give us an optimal solution. ∎

B. Alternative Proof of Theorem 4

Proof: Let $\boldsymbol{x}^*$ be an optimal RB assignment to the SINGLE SHOT problem (9). We show that there exists an ALMOST DISCRETE feasible optimal allocation $\boldsymbol{y}^*$.
For each RB k, there exists exactly one user $j^*(k)$ such that $x^*_{j^*(k)k} = 1$. Denote $R_{j^*(k)k} = R_k$ and $\gamma_{j^*(k)k} = \gamma_k$. Then substituting this optimal $\boldsymbol{x^*}$ in (9), we note that the optimal allocation $\boldsymbol{y}$ is a solution of the following LP:

$$\max \sum_k y_k/R_k \tag{21}$$

s.t.,

$$\sum_k y_k \leq C \tag{22}$$

$$0 \leq y_k \leq \gamma_k, \quad \forall k. \tag{23}$$

It is easy to see that the feasible region of the LP, given by the Eqns (22) and (23), is bounded, and hence, a finite optimal solution to the LP (21) exists. Next, recall the fundamental result that the solution of an LP is always obtained at a vertex of the feasible region [8], which is also known as the **B**asic **F**easible **S**olution (BFS).
Let κ be the number of RBs. Since there are $(\kappa+1)$ linearly independent inequalities in the constraints (22) and (23), and the dimension of the vector $\boldsymbol{y}$ is κ, it is clear that at least $(\kappa-1)$ of inequalities from (23) must be active in any **BFS**. This proves that there exists an optimal solution which is ALMOST DISCRETE.

∎

C. Proof of Lemma 6

Proof: Define the following disjoint partition of the base set $E = \bigcup_{i,k} E_{ik}$, where

$$E_{ik} = \{(i,j,k), \forall j\}.$$

Since $\boldsymbol{x}$ is feasible, it follows that at most one user may be assigned to any RB at each RU. Thus, for any $S \in \mathcal{I}$, we have $|S \cap E_{ik}| \leq 1, \forall i,k$. Hence, the proof directly follows from Example 12.8 (p. 288) of [8]. ∎

D. Proof of Lemma 7

Proof: Let S be a set of RBs and consider two RBs j,k such that $j \in S^c$ and $k \in S^c$. Then, following [9], the following inequality (24) establishes submodularity of the function $f(S)$:

$$f(S \cup \{j\}) + f(S \cup \{k\}) \geq f(S \cup \{j,k\}) + f(S). \tag{24}$$

To show the above inequality, we exploit the result in Lemma 3. Suppose, in the optimal solution corresponding to $f(S \cup j)$, the RB j was allocated a rate of r_j. Similarly, in the optimal solution to $f(S \cup k)$, the RB k was allocated a rate of x_k. And finally, in the optimal solution corresponding to $f(S \cup \{j,k\})$, the RBs j and k was allocated a rate of y_j and y_k respectively. Then, it is clear from the Lemma 3 that $r_j \geq y_j$ and $x_k \geq y_k$. Then, we may write

$$\begin{aligned} f(S \cup \{j\}) - f(S) &= r_j \delta_j \\ f(S \cup \{k\}) - f(S) &= x_k \delta_k \\ f(S \cup \{j,k\}) - f(S) &= y_j \delta_j + y_k \delta_k, \end{aligned}$$

where δ_j and δ_k denote the marginal utility of adding the RBs to the set S (this situation is equivalent to adding a new variable in the simplex method for solving the LP and the quantities δ_j and δ_k correspond to the reduced costs of the variables corresponding to the RBs j and k [21]). Using the above relations, we have

$$\begin{aligned} & \big(f(S \cup \{j\}) - f(S)\big) + \big(f(S \cup \{k\}) - f(S)\big) \\ \geq \; & f(S \cup \{j,k\}) - f(S). \end{aligned}$$

Rearranging the above, we have

$$f(S \cup \{j\}) + f(S \cup \{k\}) \geq f(S \cup \{j,k\}) + f(S),$$

and this establishes the submodularity property of the function $f(\cdot)$. ∎