Wireless Scheduling for Heterogeneous Services With Mixed Numerology in 5G Wireless Networks

Ti Ti Nguyen, Student Member, IEEE, Vu Nguyen Ha, and Long Bao Le, Senior Member, IEEE

Abstract—This letter studies the scheduling problem which determines how time-frequency resources of different numerologies can be allocated to support heterogeneous services in 5G wireless systems. Particularly, this problem aims at scheduling as many users as possible while meeting their required service delay and transmission data. To solve the underlying integer programming (IP) scheduling problem, we first transform it into an equivalent integer linear program (ILP) and then develop two algorithms, namely Resource Partitioning-based Algorithm (RPA) and Iterative Greedy Algorithm (IGA) to acquire efficient resource scheduling solutions. Numerical results show the desirable performance of the proposed algorithms with respect to the optimal solution and their complexity-performance tradeoffs.

Index Terms—5G NR, wireless scheduling, mixed numerology.

I. INTRODUCTION

The 5G wireless network is designed to support diverse applications and use cases with different requirements including the enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultra-reliable and low-latency communication (uRLLC) [1]. Toward this end, the 5G wireless standard adopts the so-called mixed numerology to enable flexible configurations and assignment for different types of physical resource blocks (PRBs) and support different wireless services [2]–[4]. Particularly, while the 4G system supports only one numerology where its PRB has the bandwidth of 180 kHz and time duration of 0.5 ms, the 5G’s PRBs can have the bandwidth equal to 2, 4, 8, 16 times of 180 kHz and the time duration equal to 2, 4, 8, 16 times smaller than 0.5 ms. The scheduling problem for resource allocation (RA) of two-dimensional (2D) time-frequency resources has been studied in several recent works [5], [6]. However, the 5G-NR mixed numerology has not been fully studied in [5] while [6] assumes that only two different numerologies exist in the system. Finally, the authors of [7] study the RA problem with mixed numerology for capacity enhancement.

In this paper, we study the scheduling problem for heterogeneous services with mixed numerology which aims to maximize the number of admitted users while meeting service latency and data transmission requirements. Two algorithms, namely Resource Partitioning-based Algorithm (RPA) and

IEEE COMMUNICATIONS LETTERS, VOL. 24, NO. 2, FEBRUARY 2020


Authorized licensed use limited to: University of Luxembourg. Downloaded on May 11,2021 at 16:17:50 UTC from IEEE Xplore. Restrictions apply.
its location is given as \( f \in [(i-1)\Delta_{\text{min}}^t : i\Delta_{\text{min}}^t) \) and \( t \in [(j-1)\Delta_{\text{min}}^t : j\Delta_{\text{min}}^t) \), for \( 1 \leq i \leq M^t \) and \( 1 \leq j \leq N^t \).

**A. Problem Formulation**

PRBs are allocated to users where the numerology is selected in advance by each user. Moreover, each RE is allocated to only one user and associated with one numerology. We represent the mapping for one particular PRB of numerology \( l \) to REs in the 2-D frequency-time resource space as follows:

\[
q_{l,m,n} = \{(i,j)|m - l \leq i \leq M + t, n - l \leq j \leq N + t\},
\]

where \( M_l = 2^l-\text{len}_1 - 1 \), \( N_l = 2^l-\text{len}_2 - 1 \). Assuming that the number of PRBs assigned to user \( k \) is not larger than \( C_k \) to maintain certain fairness among users. Then, we introduce binary variables \( x_{l,i,j,k} \) where \( y_{m,n} = 1 \) if \( q_{l,m,n} \) corresponds to the \( l^{th} \) assigned PRB of user \( k \), and \( y_{m,n} = 0 \) otherwise; \( x_{l,i,j} = 1 \) if RE (\( i,j \)) is assigned for user \( k \) in its \( l^{th} \) PRB, and \( x_{l,i,j} = 0 \) otherwise. For \( k \in K \), \( c \in C \), \( i,j,m,n \), and \( n = 1 : N_t - N_{lk}, \). We impose the following constraints to ensure non-overlapping RA:

\[
\sum_{i' = m + M_l}^{m + M_l + 1} \sum_{j' = n + N_{lk} + 1}^{n + N_{lk} + 1} x_{l,i',j',k} \geq 2^{l-1} y_{m,n}, \quad \forall k, c, m, n,
\]

\[\sum_{k \in K} x_{l,i,j,k} \leq 1, \quad \forall i,j, \quad \text{and} \quad \sum_{m = 1}^{M^t - M_{lk}} \sum_{n = 1}^{N^t - N_{lk}} c_{i,j} \leq 1 r_{m,n}^k y_{m,n}, \quad \forall k, c.
\]

Each user \( k \) wants its data chunk to be completely transmitted and the total waiting time be not larger than \( \tau_{\text{max}} \). Let \( \tau_{k,0} \) denote the initial waiting time of the data chunk of user \( k \in K \) at the beginning of the considered scheduling interval. Then, the total waiting time until the transmission instant of user \( k \) can be written as \( \tau_k = \tau_{k,0} + \tau_{k,1} \), where \( \tau_{k,1} \) is the additional waiting time before user \( k \) is served in the scheduling interval, which can be expressed as \( \tau_{k,1} = \Delta_{\text{min}}^t \min(j-1) | x_{l,i,j} = 1, \forall i,j,c \).

Our design aims to schedule as many users as possible while meeting their data demand and latency requirements. Recall that user \( k \) wishes to transmit a data chunk of \( d_k^q \) bits with the total waiting time not larger than \( \tau_{\text{max}} \). To maintain these constraints, we define a function capturing if both constraints are satisfied as \( u_k = \mathbb{1}_{d_k - d_k^q \leq \tau_{\text{max}}} \), where \( \mathbb{1}_x \) stands for the step function, i.e., \( \mathbb{1}_x = 1 \) if \( x \geq 0 \), and \( \mathbb{1}_x = 0 \) otherwise. In fact, if a scheduling solution ensures that the amount of transmitted data and the total waiting time satisfy \( d_k \geq d_k^q \) and \( \tau_k \leq \tau_{\text{max}} \), respectively, we have \( u_k = 1 \); otherwise.

\[\tau_k = 0 \text{ Then, the scheduling problem can be formulated as (Pr)}_1 \quad \max_{x,y} \sum_{k \in K} u_k \text{ s.t. } (2a), (2b), \text{ and } x, y \in \{0,1\},\]

where \( x = \{x_{l,i,j,k} \mid \forall i,j,k,c \} \) and \( y = \{y_{m,n} \mid \forall k,c,m,n \} \).

**B. Problem Transformation**

To deal with non-continuous and non-linear functions in problem \((\text{P}_1)\), we first transform this problem into a standard ILP. Specifically, we can equivalently express \( u_k \) as

\[u_k \in \{0,1\}; \quad u_k(\tau_{\text{max}} - \tau_k) \geq 0; \quad u_k(d_k - d_k^q) \geq 0. \]

Then, we introduce auxiliary variables \( x_{l,i,j,k} \)'s as \( x_{l,i,j,k} \). Using \( x_{l,i,j,k} \)'s, \((\text{P}_1)\) can be transformed into the ILP form as stated in the following proposition.

**Proposition 1:** \((\text{P}_1)\) is equivalent to the following problem

\[\text{(P}_{1\text{ILP}}) \quad \max_{x,y,z} \sum_{k \in K} u_k \text{ s.t. } (2a), (2b), \text{ and } x, y, z \in \{0,1\}, \]

\[\sum_{m = 1}^{M^t - M_{lk}} \sum_{n = 1}^{N^t - N_{lk}} c_{i,j} \geq 1 r_{m,n}^k y_{m,n} - u_k \geq 0, \quad \forall k \in K, \]

\[\Delta_{\text{min}}^t \sum_{m = 1}^{M^t - M_{lk}} \sum_{n = 1}^{N^t - N_{lk}} c_{i,j} y_{m,n}^k - u_k d_k^q \geq 0, \quad \forall k \in K, \]

\[x_{l,i,j,k} \geq u_k + y_{m,n} - 1, \quad \forall k,c,m,n, \quad \text{and } \sum_{m = 1}^{M^t - M_{lk}} \sum_{n = 1}^{N^t - N_{lk}} c_{i,j} y_{m,n}^k \leq \min\{u_k, y_{m,n}^k\}, \quad \forall k,c,m,n, \]

where \( z = \{x_{l,i,j,k} \mid \forall k,c,m,n \} \) and \( u = \{u_k, d_k^q \mid \forall k \} \).

Here, \( N_{\text{tot}} \) represents the maximum number of time slots (with size of \( \Delta_{\text{min}}^t \) seconds) that user \( k \) can wait, counting from the beginning of the scheduling interval, to meet its delay constraint which is determined as \( N_{\text{tot}} = \lfloor (\tau_{\text{max}} - \tau_{k,0})/\Delta_{\text{min}}^t \rfloor \), where \( \lfloor \cdot \rfloor \) is the floor operation.

**Proof:** \((4c) \) with \( x, y, z, u \in \{0,1\} \) are equivalent to \( x_{l,i,j,k} \). As a result, \((4a) \) and \((4b) \) in \((\text{P}_1)\) are equivalent to \((3) \). Hence, \((\text{P}_1)\) is equivalent to \((\text{P}_{1\text{ILP}})\). 

**Proposition 2:** \((\text{P}_{1\text{ILP}})\) is \(\mathcal{NP}\)-hard.

**Proof:** If the number of PRBs assigned to each user is known, the remaining problem can be formulated as a well-known “Cutting Stock Problem” (CSP), which is strongly \(\mathcal{NP}\)-hard \cite{11}. Thus, \((\text{P}_{1\text{ILP}})\) is more complex than the standard CSP since the numbers of PRBs assigned for different users should also be optimized. Therefore, \((\text{P}_{1\text{ILP}})\) must be \(\mathcal{NP}\)-hard.

**III. PROPOSED ALGORITHMS**

**A. Resource Partitioning Based Algorithm (RPA)**

We propose a low-complexity algorithm which solves \((\text{P}_{1\text{ILP}})\) by decomposing it into parallel small-scale subproblems. In fact, if we can maintain the relationship between available resources and users’ demands in each sub-problem similar to that in the original problem, then solving small-scale sub-problems can return a solution as good as the one obtained by directly solving the original problem. To realize this idea, we first divide the available bandwidth into \( M_B \) sub-bands where sub-band \( m_B \) occupies the spectrum from \((m_B - 1)M_B / M_B \) to \( m_B M_B / M_B \) such that \( m_B M_B / M_B \leq \Delta_{\text{min}} \leq M_B / M_B \Delta_{\text{min}} \). Then, the following three steps are taken in RPA: 1) perform RA on each sub-band,
Algorithm 1 Resource Partitioning Based Algorithm (RPA)

1: Initialize: Set initial value for $M_k$.
2: Partition resources into $M_k$ sub-bands and distribute users into these sub-bands as in Section III-A.
3: **Step 1**: Solve ($P_{1,RPA}$) to obtain $u_k^{SI*}$ for all $m_k$, $k \in K_m$.
4: **Step 2**: Solve ($P_{m,RPA}$) to create a contiguous region of unallocated resources between two consecutive sub-bands while still satisfying the requirements of admitted users with $u_k^{SI*} = 1$, $\forall k$.
5: **Step 3**: Solve ($P_{RPA}$) to assign unallocated resources to unadmitted users, i.e., users with $u_k^{SI*} = 0$, $\forall k$.

The key steps of RPA are illustrated in Fig. 2 and it is summarized in Algorithm 1. In **Step 1**, we randomly distribute users into sub-bands to make resource demands on different sub-bands similar. Denote the set of users associated with sub-band $m_k$ as $K_m$, and the set of REs in the frequency dimension as $\mathcal{I}_{m_k} = \{m|m = (m_k-1)M_f/M_m, \text{min}\{m|M_f/M_m, M_f\}\}$. We then solve ($P_{1,RPA}$) corresponding to each sub-band $m_k$ and the set of users $K_m$ to obtain admission decisions, denoted as $u_k^{SI*}$'s. The sub-problem for sub-band $m_k$ is named ($P_{1,RPA}$). In **Step 2**, after finding $u_k^{SI*}$'s, we re-arrange the allocated resources so that the unallocated REs from two consecutive sub-bands can be arranged close to one another and they can be combined and mapped into PRBs of certain numerology as defined in (1). The re-arrangement of unallocated REs on sub-band $m_k$ can be achieved by solving the following problem:

$$
\min_{x,y,z \in \{0,1\}} \sum_{k \in K_m} \sum_{i,j=1}^{N-I-N_k} \sum_{c=1}^{C_k} W_i x_{i,j}^{k,c} \quad \text{s.t.} \quad u_k^{SI*} = (2a), (2b), (4a), (4b), \quad \forall k \in K_m, \quad i \in \mathcal{I}_{m_k},
$$

where $\{W_i\}$ is an increasing series, e.g., $W_i = 2^i$ if the sub-band index is odd and $\{W_i\}$ is a decreasing series, e.g., $W_i = 2^{-i}$ if the sub-band index is even. It can be verified that after solving ($P_{m,RPA}$), all unallocated REs in two consecutive sub-bands will be pushed close to one another to create a contiguous resource region as large as possible.

Let $\{x_{P_m}, y_{P_m}, z_{P_m}\}$ denote the optimal solution of ($P_{m,RPA}$). In **Step 3**, we assign the resource to admitted users, $\Omega = \{(i,j) | x_{P_m}^{i,j} = 0, \forall i \in \mathcal{I}_{m_k}\}$, to the set of unadmitted users $\mathcal{K} = \{k | u_k^{SI*} = 0\}$ by solving the following problem ($P_{RPA}$):

$$
\max_{x,y,z \in \{0,1\}} \sum_{k \in \mathcal{K}} u_k \quad \text{s.t.} \quad (2a), (2b), (4a), (4b), \forall k \in \mathcal{K}, (i,j) \in \Omega.
$$

Algorithm 2 Iterative Greedy Algorithm (IGA)

1: Initialize: $d_{k,0}^0 = d_{k,0}^c = 0$, $\forall m, n = 1$, $W = 10$.
2: **repeat**
3: Compute $U_{m,n,k}$ find the largest value of $U_{m,n,k}$, and perform the corresponding PRB allocation.
4: Update different parameters after the PRB allocation as $d_{k,0}^c = c_k + 1$, assign $y_{m,n,k} = 1$, update the remaining required data $d_{k,0}^c = d_{k,0}^c - r_{m,n,k} \Delta f_k^c$, and $W_{m,n,k} = W_{m,n,k} = W, \forall n = 1 : N^t$.
5: Drop all overlapped PRBs $q_{k}^{k}, q_{m,n,k}$
6: **until** $U_{m,n,k} = 0, \forall m, n, k$

B. Iterative Greedy Algorithm (IGA)

We propose another fast iterative algorithm where we greedily assign resources to users based on an assignment weight which depends on the requirements of the underlying user, the amount of data transmitted and the achieved latency if the underlying PRB is assigned to the user. In each iteration, we calculate the assignment weight for each pair of an available PRB and a user based on which the resource assignment is performed for the PRB-user pair achieving the largest weight. After that, the available PRBs and the weights of all possible PRB-user pairs are updated to prepare for further resource assignment in the next iteration. This process is repeated until there is no more available PRB or unsatisfied user.

We now define the resource assignment weight for a particular PRB-user-resource pair as follows: $U_{m,n,k} = d_{k,0}^{c_k} N_k^{m,n} \mathcal{N}_{k}^{m,n}$ if $c_k \leq C_k$, $d_{k,0}^{c_k} > 0$, and $U_{m,n,k} = 0$, otherwise, where $d_{k,0}^{c_k}$ is the remaining required data amount in each iteration, which is equal to $d_{k,0}^{c_k}$ in the first iteration, $c_k$ is the current total PRBs assigned to user $k$, and $N_k$ is the set of REs in the time domain, which is defined as $N_k = \{n | n \leq N^t\}$ if $\sum_{n=1}^{N^t} M^t - M_k \sum_{C_k=1}^{C_k} y_{n,c}^{k} = 0$, and $N_k = \{n | n \leq N^t\}$, otherwise, and $W_{m,n,k}$ is a matrix used to mitigate the resource fragmentation in the allocation process, which is updated in each iteration. The unit matrix is initially assigned to $W_{m,n,k}$. In particular, $U_{m,n,k}$ is chosen based on the following criteria: 1) Users with smaller required data chunk receive higher scheduling priorities; 2) If a user is not admitted yet, a PRB at the time location closer to the slot corresponding to the allowed maximum waiting time is more prioritized; 3) Admitted users are allocated resources until their requirements are completely satisfied; and 4) The resource fragmentation is prevented to ease future PRB allocations. The IGA is summarized in Algorithm 2. In each iteration of IGA, we need to compute the resource assignment weights $U_{m,n,k}$ for all available $m, n, k$, determine the largest $U_{m,n,k}$ to perform RA, update different parameters, and drop all overlapped PRBs to the assigned block. The worst-case complexity of each iteration is $O(M^t N^t K)$. Let $N^\text{iter}$ be the number of iterations, which is upper bounded by $M^t N^t |\mathcal{L}|$, the overall worst-case complexity of IGA is $O(N^\text{iter} M^t N^t |\mathcal{L}| K)$.

IV. NUMERICAL RESULTS

We consider a wireless system with pedestrian and high moving users in a cell with radius of 500 meters. The channel path-loss $\beta_k$ (dB) = 128.1 + 37.6 log_{10}(\gamma_k)$ where $\gamma_k$ is the
distance between user $k$ and the BS (in km). For small-scale channel fading, ITU pedestrian-B channel parameters with Doppler shift of 50 Hz and ITU Vehicular-A channel parameters with Doppler shift of 500 Hz are used for pedestrian users and high moving users, respectively. We set three user groups A, B, and C and the numbers of users in these group are $|K/3|$, $|K/3|$ and $K-2|K/3|$, respectively. Specifically, group A adopts numerology 0 corresponding to high moving users with large data demand, group C uses numerology 2 corresponding to high moving users requiring low waiting time, and group B employs numerology 1 corresponding to pedestrian users with average requirements on the transmission data and waiting time. The transmission rate is calculated according to Shannon’s capacity where the ratio of transmit power per Hz to noise power density is set equal to $2.8 \times 10^5$. The required data chunks $d_{rk}^q$ over the interval $T$ of 1 ms for users in groups A, B, and C are set randomly in $[500 – 2000]$ (bits), $[500 – 1000]$ (bits), and $[180 – 500]$ (bits), respectively, and $C_k$ is set equal to 10. Besides, $N_k^{rq}$ defined in Proposition 1 is set equal to 8 for users in group A and randomly in [3-6] and [1-4] for users in groups B and C, respectively. All numerical results are averaged by obtaining the results over 50 random realizations.

We show the relative gap in Fig. 3-(a) which is calculated as $\left(\sum_k u_k^G - \sum_k u_k^{RPA|IGA}\right) \times 100\% / \sum_k u_k^G$ for $M^f = 32$ and $K = 20$ where $u_k^{RPA|IGA}$ and $u_k^G$ represent the objective values for user $k$ obtained by using RPA/IGA/LP+LD [7] and the CVX-Gurobi solver, respectively. Fig. 3-(b) shows the ratio between the average execution time (ET) of the RPA/IGA and that required to solve $(P_{1\text{LP}}^{ILP})$ by the CVX-Gurobi solver (ET-G). The “LP+LD” algorithm proposed in [7] includes two loops: an outer loop to assign PRBs to users based on the utility matrix for all PRB-user pairs, and an inner loop to determine the utility matrix. The figure shows that our proposed algorithms outperform the “LP+LD” algorithm. Besides, as shown in this figure, the relative gap due to RPA increases when the number of sub-bands $M_B$ increases. This is because larger $M_B$ reduces RA flexibility in each sub-band and thus resource utilization efficiency. However, the execution time of the RPA can be reduced significantly when $M_B$ becomes larger. In contrast, IGA always explores good resource-user pairs for efficient resource utilization and IGA is not affected by $M_B$.

Fig. 4-(a) shows the admission ratio achieved by IGA with different values of $M^f$ where this ratio is equal to the number of admitted users divided by the total number of users. As can be seen, the admission ratio decreases with the increasing number of requesting users as expected. Moreover, the proposed IGA can admit all users when there are sufficient network resources. We study the interactions between two user groups with different requirements and $M^f = 64$ in Fig. 4-(b). Specifically, group one has varying data transmission demand with maximum waiting time of 1ms while group two requires smaller waiting time compared to group one and each user has a data chunk of $d_{rk}^q = 250$ bits. In addition, there are 40 users in group one selecting numerology 0 and 40 users in group two using numerology 2 with the same maximum waiting time. For group two, we consider three different values of $N_k^{rq}$, which are 1, 2, 3. It can be observed that the higher the data transmission demand per user of group one, the lower the admission ratio that can be achieved by IGA. Also, the improvement in the admission ratio when $N_k^{rq}$ increases from 1 to 2 is considerably greater than that when $N_k^{rq}$ increases from 2 to 3. This implies that strict delay requirement may have very negative impact on the system performance.

V. CONCLUSION

We have proposed two low-complexity algorithms to tackle the scheduling problem for the 5G wireless system supporting heterogeneous services. Numerical results have revealed the desirable performance and complexity for the proposed algorithms.

REFERENCES