Optimal Resource Allocation for Multi-User OFDMA-URLLC MEC Systems

WALID R. GHANEM (Student Member, IEEE), VAHID JAMALI (Member, IEEE), AND ROBERT SCHOFER (Fellow, IEEE)

1Institute for Digital Communications, Friedrich–Alexander-University Erlangen–Nürnberg, 91058 Erlangen, Germany
2Department of Electrical Engineering and Information Technology, Technical University of Darmstadt, 64283 Darmstadt, Germany

ABSTRACT In this paper, we study resource allocation algorithm design for multi-user orthogonal frequency division multiple access (OFDMA) ultra-reliable low latency communication (URLLC) in mobile edge computing (MEC) systems. To meet the stringent end-to-end delay and reliability requirements of URLLC MEC systems, we employ joint uplink-downlink resource allocation and finite blocklength transmission. Furthermore, we propose a partial time overlap between the uplink and downlink frames to minimize the end-to-end delay, which introduces a new time causality constraint. The proposed resource allocation algorithm is formulated as an optimization problem for minimization of the total weighted power consumption of the network under a constraint on the number of URLLC user bits computed within the maximum allowable computation time, i.e., the end-to-end delay of a computation task of each user. Despite the non-convexity and the complicated structure of the formulated optimization problem, we develop a globally optimal solution using a branch-and-bound approach based on discrete monotonic optimization theory. The branch-and-bound algorithm minimizes an upper bound on the total power consumption until convergence to the globally optimal value. Furthermore, to strike a balance between computational complexity and performance, we propose two efficient suboptimal algorithms. For the first suboptimal scheme, the optimization problem is reformulated in the canonical form of difference of convex programming. Then, successive convex approximation (SCA) is used to determine a locally optimal solution. For the second suboptimal scheme, we use a high signal-to-noise ratio approximation for the channel dispersion. Then, via novel transformations, we convert the non-convex quality-of-service constraints of the original problem into equivalent second-order-cone constraints. Our simulation results reveal that the proposed resource allocation algorithm design facilitates URLLC in MEC systems, and yields significant power savings compared to three baseline schemes. Moreover, our simulation results show that the proposed suboptimal algorithms offer different trade-offs between performance and complexity and attain an excellent performance at comparatively low complexity.

INDEX TERMS Branch-and-bound, finite blocklength transmission, mobile edge computing, successive convex approximation, ultra-reliable low latency communication.

I. INTRODUCTION

FUTURE wireless communication networks target several objectives including high data rates, reduced latency, and massive device connectivity. One important objective is to facilitate ultra-reliable low latency communication (URLLC). URLLC is crucial for mission-critical applications such as remote surgery, factory automation, autonomous driving, tactile Internet, and augmented reality to enable real-time machine-to-machine and human-to-machine interaction [1], [2], [3]. URLLC imposes strict quality-of-service (QoS) constraints including a low packet error probability (e.g., $10^{-6}$) and a very low latency (e.g., 1 ms). This makes the resource allocation problem particularly challenging.

1. This paper was presented in part at IEEE GLOBECOM 2020 [1].
design of URLLC systems challenging as ultra-high reliability and low-latency are two conflicting design objectives. In particular, low latency mandates finite blocklength transmission (FBT), which in turn severely compromises the reliability [4]. Furthermore, FBT makes the design of resource allocation challenging due to the non-convexity of the corresponding rate expression in terms of the transmission power, blocklength, and decoding error probability [5].

Recently, significant attention has been devoted to studying and developing resource allocation algorithms for URLLC. In particular, resource allocation for orthogonal frequency division multiple access (OFDMA)-URLLC systems was studied in [6], [7], [8], [9]. In [10], [11], resource allocation for secure URLLC was investigated. Furthermore, the application of multiple-antenna techniques to URLLC systems was investigated in [6], [12], [13]. However, the resource allocation schemes in [6], [7], [8], [10], [11], [12], [13], [14] focused only on communication while computation was not considered. Nevertheless, devices in mission-critical applications are expected to generate tasks that require computation within a given time. This motivates the investigation of resource allocation algorithm design for efficient computation in URLLC systems. A promising solution to enable efficient and fast computation for URLLC devices is mobile edge computing (MEC). MEC can enhance the battery lifetime and reduces the power consumption of users with delay-sensitive computation tasks [15]. By offloading these tasks to nearby MEC servers, the power consumption and computation time at the local users can be considerably reduced at the expense of the power required for data transmission for offloading [15]. In general, there are two offloading methods for computation tasks, namely, partial and binary offloading [16]. For partial offloading, the users partition their computation task into two parts where one of these parts is offloaded to the MEC server for computing [17]. Meanwhile, for binary offloading, the computation cannot be partitioned and has to be either completely offloaded or computed locally. For both offloading modes, careful resource allocation is paramount to ensure the efficient use of the available resources (e.g., power and bandwidth) while guaranteeing a maximum delay for task computation.

Resource optimization and task offloading for MEC-enabled Internet of things (IoT) systems has been extensively studied in the literature [18], [19], [20], [21]. In particular, the authors of [18], [20] investigated energy-efficient resource allocation for MEC, while computation rate maximization was targeted in [19]. Resource allocation design for OFDMA MEC systems was studied in [22]. Unmanned aerial vehicle (UAV) assisted MEC was studied in [23], [24], [25], [26], [27]. Moreover, machine learning based methods for task offloading in MEC networks have been employed in [28], [29]. In particular, a deep reinforcement learning algorithm for online computation offloading in wireless powered MEC systems based on the memory replay technique was studied in [30]. Furthermore, in [31], the authors investigated joint communication and computation resource allocation for non-orthogonal multiple access (NOMA) assisted MEC systems. However, the aforementioned resource allocation schemes [18], [19], [20], [22], [25], [26], [27], [30], [31] were based on Shannon’s capacity formula for the additive white Gaussian noise (AWGN) channel. Since URLLC systems exploit a short frame structure and a small packet size to reduce latency, the relation between the achievable rate, decoding error probability, and transmission delay cannot be captured by Shannon’s capacity formula which assumes infinite block length and zero error probability [32]. If Shannon’s capacity formula is utilized for resource allocation design for URLLC systems, the latency will be underestimated and the reliability will be overestimated, and as a result, the QoS requirements of the users cannot be met.

To overcome this issue, some recent works considered FBT [5] for resource allocation algorithm design for URLLC MEC systems. In particular, the authors in [33] studied binary offloading in single-carrier TDMA systems. However, single-carrier systems suffer from poor spectrum utilization and require complex equalization at the receiver. In [34], the authors investigated the minimization of the normalized energy consumption of OFDMA-URLLC MEC systems. However, the algorithm proposed in [34] assumes that the channel gains of different sub-carriers are identical which may not be a realistic assumption for broadband wireless channels. Moreover, the resource allocation algorithms proposed in [34] are based on a simplified version of the general expression for the achievable FBT rate [5], which is tight only at medium-to-high values of signal-to-noise ratio (SNR). Furthermore, the existing MEC designs, such as [18], [25], do not take into account the size of the computation result of the tasks and do not consider the communication resources consumed for downloading of the processed data by the users. However, the size of the processed data can be large for applications such as augmented reality.

We note that most resource allocation algorithms proposed for URLLC systems in the literature, such as [10], [33], [34], [35], [36], are strictly suboptimal. In particular, the algorithms developed in [10], [35] are based on block coordinate descent techniques, while those in [36], [37] employ successive convex approximation (SCA). As a result, the performance of these resource allocation algorithms cannot be guaranteed because the gap between the optimal and suboptimal solutions is not known. In our recent work [6], we proposed a global optimal algorithm based on the polyblock outer approximation method using monotonic optimization. However, the polyblock algorithm may suffer from slow convergence for large problem sizes. To overcome this problem, in this paper, a branch-and-bound algorithm is proposed. Different from the general branch-and-bound algorithms developed for non-convex problems, e.g., [38], the proposed branch-and-bound algorithm exploits the monotonicity of the problem to reduce the search space for faster convergence [39]. Moreover, the proposed branch-and-bound
algorithm eliminates the need for internal solvers, such as CVX, which significantly reduces the time for determining the optimal solution.

In this paper, we study the optimal joint uplink-downlink resource allocation design for OFDMA-URLLC MEC systems with FBT. The main contributions of this paper are summarized in the following:

- We propose a novel joint uplink-downlink resource allocation algorithm design for multi-user OFDMA-URLLC MEC systems with FBT, and study the impact of FBT on the radio resource management for binary offloading. To reduce the end-to-end delay of uplink and downlink transmission, while efficiently exploiting the available spectrum, we propose a partial time overlap between the uplink and downlink frames and introduce corresponding causality constraints.\(^2\) The resource allocation algorithm design is formulated as an optimization problem for the minimization of the total weighted power consumed by the base station (BS) and the users subject to QoS constraints for the URLLC users. The QoS constraints include the required number of bits computed within a maximum allowable time, i.e., the maximum end-to-end delay of each user.

- The formulated optimization problem is a non-convex mixed-integer problem and difficult to solve. Thus, we transform the problem into the canonical form of a discrete monotonic optimization problem. This reformulation allows the application of the branch-and-bound algorithm to find the global optimal solution. The proposed branch-and-bound algorithm searches for a global optimal solution by successively partitioning the non-convex feasible region and using bounds on the objective function to discard inferior partition elements. Different from the generic branch-and-bound algorithms in the literature [38], this algorithm exploits the monotonicity of the problem at hand to reduce the search space and obtain faster convergence.

- To strike a balance between computational complexity and performance, we develop two efficient low-complexity suboptimal algorithms based on SCA. For the first suboptimal scheme, we transform the original optimization problem into the canonical form of difference-of-convex programming. Subsequently, Taylor series approximation is utilized to obtain a local optimal solution. For the second suboptimal scheme, we use a high SNR approximation for the channel dispersion and show that the non-convex QoS constraints can be converted into equivalent second-order-cone (SOC) constraints based on novel transformations.

- Our simulations show that the proposed suboptimal algorithms offer different trade-offs between complexity and performance. Furthermore, the proposed algorithms achieve significant performance gains compared to three baseline schemes.

We note that this paper expands the corresponding conference version [1] in several directions. First, the formulated optimization problem targets joint local computing and edge offloading, while only edge offloading was considered in [1]. Second, we derive the optimal resource allocation policy for OFDMA-URLLC MEC systems, whereas only a suboptimal algorithm was provided in [1]. Thirdly, we propose a second suboptimal algorithm to further reduce the complexity of the suboptimal scheme proposed in [1].

The remainder of this paper is organized as follows. In Section II, we present the considered system and channel models. In Section III, the proposed resource allocation problem is formulated. In Section IV, the optimal resource allocation algorithm is derived, whereas low-complexity suboptimal algorithms are provided in Section V. In Section VI, the performance of the proposed schemes is evaluated via computer simulations, and finally conclusions are drawn in Section VII.

**Notation:** Lower-case letters \( x \) refer to scalar numbers, and bold lower-case letters \( \mathbf{x} \) represent vectors. \((\cdot)^T\) denotes the transpose operator. \( \mathbb{R}^{N \times 1} \) represents the set of all \( N \times 1 \) vectors with real valued entries. The circularly symmetric complex Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \) is denoted by \( \mathcal{CN}(\mu, \sigma^2) \), \( \sim \) stands for “distributed as”, and \( \mathcal{E}\{\cdot\} \) denotes statistical expectation. \( \nabla_x f(\mathbf{x}) \) denotes the gradient vector of function \( f(\mathbf{x}) \) and its elements are the partial derivatives of \( f(\mathbf{x}) \). For any two vectors \( \mathbf{x}, \mathbf{y} \in \mathbb{R}^{+} \), \( \mathbf{x} \preceq \mathbf{y} \) means \( x_i \leq y_i, \forall i \), where \( x_i \) and \( y_i \) are the \( i \)-th elements of \( \mathbf{x} \) and \( \mathbf{y} \), respectively. \( \mathbf{x}^+ \) denotes the optimal value of an optimization variable \( \mathbf{x} \).

**II. SYSTEM AND CHANNEL MODELS**

In this section, we present the system and channel models for the considered OFDMA-URLLC MEC system.

**A. SYSTEM MODEL**

We consider a single-cell multi-user MEC system which comprises a BS and \( K \) URLLC users indexed by \( k = \{1, \ldots, K\} \), cf. Fig. 1. All transceivers are equipped with single antennas. The system employs frequency division duplex (FDD).\(^3\) Thereby, the total bandwidth \( W \) is divided into two bands for uplink and downlink transmission having bandwidths \( W^u \) and \( W^d \), respectively. The bandwidths for uplink and downlink transmission are further divided into \( M^u \) and \( M^d \) orthogonal sub-carriers indexed by \( m^u = \{1, \ldots, M^u\} \) and \( m^d = \{1, \ldots, M^d\} \), respectively. The bandwidth of each sub-carrier is \( B_s \), leading to a symbol duration of \( T_s = \frac{1}{B_s} \). The uplink and downlink frames are divided into \( N^u \) time slots indexed by \( n^u = \{1, \ldots, N^u\} \) and \( N^d \) time slots indexed by \( n^d = \{1, \ldots, N^d\} \), respectively. Moreover, each time slot contains one orthogonal frequency division multiplexing.
OFDM) symbol. Each user has one computation task \((B_k, D_k)\) that needs to be processed, where \(B_k\) is the task size in bits and \(D_k\) is the deadline for computation in time slots. For task offloading, the user sends the task in the uplink and the edge server computes the task and sends the results back to the user in the downlink. There is an offset of \(\tau\) time slots between uplink and downlink transmission. Thus, uplink and downlink transmission overlap in \(\hat{O} = N^u - \tau\) time slots, and thus, the total system delay is reduced by \(\hat{O}\) time slots compared to the case without overlap. The value of \(\tau\) is a design parameter. On the one hand, if \(\tau\) is chosen too small, the users’ tasks may have not yet been computed when the downlink frame ends and hence the downlink resource is wasted. On the other hand, if \(\tau\) is chosen too large, the computed bits at the BS have to wait before being transmitted to the users, which increases the end-to-end delay, see Fig. 1. By taking advantage of the overlap between uplink and downlink transmission, users with different delay requirements can be supported. For example, users requiring low latency are assigned resource elements at the beginning of the uplink and downlink frames, cf. green resource elements in Fig. 1. The maximum transmit power of the BS is \(P_{\text{max}}\), while the maximum transmit power of user \(k\) in the uplink is \(P_{k,\text{max}}\).

In order to facilitate the presentation, in the following, we use superscript \(j \in \{u, d\}\) to denote uplink \(u\) and downlink \(d\).

Remark 1: We note that the time and power consumed for channel estimation and resource allocation are constant and do not affect the proposed resource allocation algorithm. For simplicity of illustration, they are neglected in this paper. Furthermore, perfect channel state information (CSI) is assumed to be available at the BS for resource allocation design to obtain a performance upper bound for OFDMA-URLLC MEC systems.

B. UPLINK AND DOWNLINK CHANNEL MODELS

In the following, we introduce the uplink and downlink channel models for the considered OFDMA-URLLC MEC system. We assume that the coherence time exceeds the frame duration. Therefore, for all users, the channel gains of all sub-carriers are constant during uplink and downlink transmission, i.e., for \(N^u\) and \(N^d\). In the uplink, the signal received at the BS from user \(k\) on sub-carrier \(m^u\) in time slot \(n^u\) is given as follows:

\[
\gamma^u_k[m^u, n^u] = h^u_k[m^u]x^u_k[m^u, n^u] + z^u_{BS}[m^u, n^u],
\]

where \(x^u_k[m^u, n^u]\) denotes the symbol transmitted by user \(k\) on sub-carrier \(m^u\) in time slot \(n^u\) to the BS. Moreover, \(z^u_{BS}[m^u, n^u] \sim \mathcal{CN}(0, \sigma^2)\) denotes the noise on sub-carrier \(m^u\) in time slot \(n^u\) at the BS, and \(h^u_k[m^u]\) represents the complex channel coefficient between user \(k\) and the BS on sub-carrier \(m^u\). For future reference, we define the SNR of user \(k\)’s signal at the input of the BS’s receiver on sub-carrier \(m^u\) in time slot \(n^u\) as follows:

\[
\gamma^u_k[m^u, n^u] = \frac{E_s^u|m^u, n^u|}{\sigma^2},
\]

where \(E_s^u[m^u, n^u] = E(|x^u_k[m^u, n^u]|^2)\) is the uplink transmit power of user \(k\) on sub-carrier \(m^u\) in time slot \(n^u\), and \(g^u_k[m^u] = \frac{|h^u_k[m^u]|^2}{\sigma^2}\). A similar channel model is assumed for downlink transmission and the corresponding SNR at user \(k\) on sub-carrier \(m^d\) in time slot \(n^d\) is denoted by \(\gamma^d_k[m^d, n^d]\).

C. ACHIEVABLE RATE FOR FBT

Shannon’s capacity theorem, on which most conventional resource allocation designs are based, applies to the asymptotic case where the packet length approaches infinity and the decoding error probability goes to zero [32]. Thus, it cannot be used for resource allocation design for URLLC systems, as URLLC systems have to employ short packets to achieve low latency, which makes decoding errors unavoidable. For the performance evaluation of FBT, the so-called normal approximation for short packet transmission was developed in [4]. For AWGN channels, the maximum number of bits \(\Psi\) conveyed in a packet comprising \(L_p\) symbols can be approximated as follows [4, eq. (4.277)], [40, Fig. 1]:

\[
\Psi = \sum_{l=1}^{L_p} \log_2(1 + \gamma[l]) - aQ^{-1}(\epsilon) \sqrt{\sum_{l=1}^{L_p} v[l]},
\]

4. Without loss of generality, we assume that the noise processes at all receivers have identical variances.
where $\epsilon$ is the decoding packet error probability, and $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function with $Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp(-\frac{t^2}{2})dt$. $\nu[l] = 1 - (1 + \gamma[l])^{-2}$ and $\gamma[l]$ are the channel dispersion [4] and the SNR of the $l$-th symbol, respectively, and $a = \log_2(\epsilon)$.

In this paper, we base the joint uplink-downlink resource allocation algorithm design for OFDMA-URLLC MEC systems on (3). By allocating several resource elements from the available resources to a given user, the number of offloaded and downloaded bits of the user can be adjusted.

III. PROBLEM FORMULATION
In this section, we explain the offloading and downloading process and introduce the QoS requirements of the OFDMA-URLLC MEC users. Moreover, we formulate the proposed resource allocation algorithm design as an optimization problem.

A. COMPUTING MODES
In this section, we discuss the different computing modes of the users. First, we elaborate on the local computing at the users. Then, we explain the steps required for offloading to the edge server.

1) Local Computing Mode: According to [41][42, eq. (1)], the power consumption of the central processing unit (CPU) comprises the dynamic power, short circuit power, and leakage power where the dynamic power is much larger than the other two. As a result, similar to [42], we only take into account the dynamic power consumed during local execution. According to [41], [42], [43], the total energy required for computing a task of length $B_k$ bits at user $k$ is given by:

$$E_k = \kappa c_k B_k t_k^2,$$  \hspace{1cm} (4)

where $f_k$ denotes the CPU frequency of the $k$-th user, $\kappa$ is the effective switched capacitance which depends on the chip architecture and is assumed to be identical for all users, $c_k$ is the number of cycles required for processing of one bit which depends on the type of application and the CPU architecture [43]. A user can reduce its total energy consumption by reducing the CPU frequency. However, the task computing latency also depends on the frequency and is given as follows:

$$t_k = \frac{c_k B_k}{f_k}.$$  \hspace{1cm} (5)

Combining (4) and (5), the local power consumption at user $k$ is given as follows:

$$P_k^l = \kappa f_k^3.$$  \hspace{1cm} (6)

A user can adjust its CPU frequency to minimize its local power consumption subject to a required task computing latency. Alternatively, considering the limited capability of its CPU, a user may prefer to offload its task to the edge server instead. This process is explained in the following.

2) Offloading and Downloading: The edge computing process is performed as follows. First, the user offloads its data to the edge server in the uplink. Subsequently, the edge server processes this data and sends the results back in the downlink to the user. Thus, uplink and downlink transmission should satisfy the following constraints:

$$C1 : \Psi_k^u(s_k, p_k^u) \geq (1 - \alpha_k) B_k, \forall k,$$  \hspace{1cm} (7)

$$C2 : \Psi_k^d(s_k, p_k^d) \geq (1 - \alpha_k) \Gamma_k B_k, \forall k,$$  \hspace{1cm} (8)

where

$$\Psi_k^u(s_k, p_k^u) = F_k^u(s_k, p_k^u) - \nu_k P_k^u,$$  \hspace{1cm} (9)

and

$$F_k^u(s_k, p_k^u) = \sum_{m' = 1}^{M} \sum_{n' = 1}^{N} s_k[m', n'] \log_2(1 + \gamma_k[m', n']), \forall j,$$  \hspace{1cm} (10)

$$\nu_k P_k^u = a Q^{-1}(\epsilon_k) \sum_{m' = 1}^{M} \sum_{n' = 1}^{N} s_k[m', n'] v_k[m', n'], \forall j.$$  \hspace{1cm} (11)

Here, $s_k[m', n'] = \{0, 1\}$, $\forall m', n', k, \forall j$, are the sub-carrier assignment indicators. If sub-carrier $m'$ is assigned to user $k$ in time slot $n'$, we have $s_k[m', n'] = 1$, otherwise $s_k[m', n'] = 0$. Furthermore, we assume that each sub-carrier is allocated to at most one user to avoid multiple access interference. $s_k^l$ and $p_k^l$ are the collections of optimization variables $s_k[m', n'], \forall m', n', l$, and $p_k^l[m', n'], \forall m', n', l, j$, respectively, and $v_k[m', n'] = 1 - (1 + \gamma_k[m', n'])^{-2}$. Constraints C1 and C2 guarantee the transmission of $(1 - \alpha_k) B_k$ bits in the uplink and $\Gamma_k(1 - \alpha_k) B_k$ bits in the downlink for user $k$, respectively, where parameter $\Gamma_k, \forall k$, specifies the ratio of the size of the computing result and the size of the offloaded task. The value of $\Gamma_k$ depends on the application type, e.g., $\Gamma_k > 1$ for augmented reality applications [22]. Moreover, $\alpha_k = \{0, 1\}$ is the binary mode selection variable, where $\alpha_k = 1$ for local computing and $\alpha_k = 0$ for edge computing offloading.

B. CAUSALITY AND DELAY
In the following, we explain the causality and delay constraints in the proposed OFDMA-URLLC MEC system.

1) Causality: Downlink transmission cannot start for a given user before all data of this user has been received at the BS via the uplink. Furthermore, according to Fig. 1, uplink and downlink transmission overlap in time slot $n'' = \tau + o$ or equivalently $n_d = o, \forall o = \{1, \ldots, O\}$. For the downlink, we

5. We note that partial offloading can be realized by relaxing $\alpha_k, \forall k$, such that it can assume any value in the interval $[0, 1]$. In this case, the first part of user $k$’s task, $\alpha_k B_k$, will be processed locally, while the second part, $(1 - \alpha_k) B_k$, will be offloaded to the edge server. In this paper, we focus on the binary offloading since the tasks in URLLC systems are expected to have small sizes, such that further dividing them may not justify the associated overhead.

VOLUME 3, 2022 2009
need to ensure that for each user $k$, if overlapping time slot $n^u = \tau + o$ is allocated to the uplink, no overlapping time slot with $n^d \leq o$ is allocated to the downlink. Exploiting the binary nature of variables $s^u_k[m^u, n^u]$ and $s^d_k[m^d, n^d]$, this condition can be imposed by the following set of linear inequality constraints:

$$
C_3 : s^u_k[m^u, \tau + o] + s^d_k[m^d, n^d] \leq 1, \forall k, \forall m^u, \forall n^d, \forall o = \{1, \cdots, O\}, \forall n^d = \{1, \ldots, o\}.
$$

As can be seen from (12), if user $k$ uses sub-carrier $m^u$ in uplink time slot $n^u = \tau + o$, then the downlink resources at and before time slot $n^d = o$ will be forced to be zero, i.e., no data is sent to user $k$.

2) Delay: The total latency of the computing task for user $k$, $D_k$, can be constrained by requiring the downlink transmission to be finished by time slot $D_k - \tau$ as follows:

$$
C_4 : s^d_k[m^d, n^d] = 0, \forall n^d \geq D_k - \tau, \forall k.
$$

Note that the values of $D_k$ and $\tau$ are assumed to be known for resource allocation. Moreover, as can be seen from (13), by adjusting the values of $D_k, \forall k$, different delay requirements may be improved for different users. For example, for the green user (in Fig. 1b), the total permissible delay is $D_k = 5$ time slots. If this is achieved within the available resource elements, other users with more relaxed latency requirements can use the remaining resource elements at the beginning, the middle, and the end of the frame.

C. TOTAL SYSTEM POWER CONSUMPTION

The total system power consumption includes the power consumption of the users and the BS. The power consumption of user $k$ is given as follows [19], [44], [45]:

$$
P_k = k f^3 \delta_k + \delta_k \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} s^u_k[m^u, n^u] P^u_k[m^u, n^u]
+ (1 - \alpha_k) P_{CIR},
$$

(14)

where the first term in (14) accounts for the local computation power consumption in case of local computing, the second term accounts for the power consumed for offloading transmission, and the third term accounts for the constant circuit power consumption during offloading. To model the inefficiency of the power amplifiers of the users, we introduce the multiplicative constant, $\delta_k \geq 1$, for the power radiated by the transmitter in (14) which takes into account the joint effect of the drain efficiency and the backoff of the power amplifier [46]. Note that, as can be seen from C1 and C2, when $\alpha_k = 1$, the required offloaded and downloaded data is zero, and hence, in this case, since we minimize the total power consumption, the power allocated for uplink transmission, $P^u_k[m^u, n^u]$, will be zero $\forall m^u, \forall n^u$. On the other hand, for offloading, i.e., $\alpha_k = 0$, the optimization problem formulated in the next subsection will ensure that the power consumption for local computing will be zero. Hence, there is no need to explicitly multiply the first and second term in (14) by $\alpha_k$ and $(1 - \alpha_k)$ to ensure that the terms are zero for offloading and local computing, respectively. Moreover, since in practice the BS does not only serve the MEC users considered for resource allocation but also non-MEC users, the BS circuit and computation power consumption are not considered for optimization. Thus, the relevant weighted system power consumption is modelled as follows:

$$
\Phi = \sum_{k=1}^{K} w_k P_k
+ \delta_{BS} \sum_{k=1}^{K} \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} s^d_k[m^d, n^d] P^d_k[m^d, n^d],
$$

(15)

where the second term in (15) represents the power consumption of the BS for downlink transmission and $\delta_{BS} \geq 1$ accounts for the inefficiency of the BS power amplifier. Moreover, $w_k \geq 1, \forall k$, are weights that allow the prioritization of the users’ power consumption compared to the BS’s power consumption.

D. OPTIMIZATION PROBLEM FORMULATION

In the following, we formulate the resource allocation problem with the goal to minimize the total weighted network power consumption, while satisfying the latency requirements of the users’ computing tasks. In particular, we optimize the uplink and downlink transmit powers, the uplink and downlink sub-carrier assignment, the CPU frequency of the local CPUs, and the mode selection of each user. To this end, the optimization problem is formulated as follows:

$$
\begin{align*}
\text{minimize} & \quad f, s^u, s^d, P^u, P^d \\
\text{s.t.} & \quad C_1 - C_4, C_5: \sum_{k=1}^{K} s^u_k[m^u, n^u] \leq 1, \forall m^u, \forall n^u, \\
& \quad C_6: s^u_k[m^u, n^u] \in [0, 1], \forall k, m^u, \forall n^u, \\
& \quad C_7: s^d_k[m^d, n^d] \leq 1, \forall m^d, \forall n^d, \\
& \quad C_8: s^d_k[m^d, n^d] \in [0, 1], \forall k, m^d, \forall n^d, \\
& \quad C_9: \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} s^u_k[m^u, n^u] P^u_k[m^u, n^u] \leq P_{k,\text{max}}, \forall k, \\
& \quad C_{10}: P^u_k[m^u, n^u] \geq 0, \forall k, m^u, \forall n^u, \\
& \quad C_{11}: \sum_{k=1}^{K} \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} s^d_k[m^d, n^d] P^d_k[m^d, n^d] \leq P_{\text{max}}, \\
& \quad C_{12}: P^d_k[m^d, n^d] \geq 0, \forall k, m^d, \forall n^d, \\
& \quad C_{13}: c_{\text{CAS}} B_k \leq T_{f_k} D_k, \forall k, \\
& \quad C_{14}: \alpha_k \in [0, 1], \forall k, \quad C_{15}: f_k \leq f_{\text{max}}, \forall k.
\end{align*}
$$

(16)
Here,  \( f, s^u, p^u, s^d, p^d \), and  \( \alpha \) are the collections of optimization variables  \( f_k, \forall k, s^u_k, \forall k, p^u_k, \forall k, s^d_k, \forall k, p^d_k, \forall k \), and  \( \alpha_k, \forall k \), respectively.

In (16), constraints C1 and C2 guarantee the transmission of the required number of bits from user  \( k \) to the BS in the uplink and from the BS to user  \( k \) in the downlink, respectively, if the user offloads the task, i.e.,  \( \alpha_k = 0 \). Constraint C3 is the uplink-downlink causality constraint and constraint C4 ensures that user  \( k \) is served such that its task meets the associated delay requirement. Constraints C5 and C6 for the uplink and constraints C7 and C8 in the downlink are imposed to ensure that each sub-carrier in a given time slot is allocated to at most one user. Constraints C9 and C11 are the total transmit power constraints of user  \( k \) and the BS, respectively. Constraints C10 and C12 are the non-negative transmit power constraints. Constraint C13 ensures that the maximum allowed delay for local computing is not exceed when  \( \alpha_k = 1 \). Constraint C14 is the mode selection constraint. Finally, constraint C15 limits the CPU frequency of the local CPUs to  \( f_{\text{max}} \).

Remark 2: Resource allocation algorithm design for conventional MEC systems is typically based on Shannon’s capacity formula, i.e.,  \( V_c^k(s^u_k, p^u_k) \) and  \( V_d^k(s^d_k, p^d_k) \) in C1 and C2 are absent. The presence of  \( V_c^k(s^u_k, p^u_k) \) and  \( V_d^k(s^d_k, p^d_k) \) makes optimization problem (16) significantly more difficult to solve but is essential for capturing the characteristics of OFDMA-URLLC MEC systems.

Problem (16) is a mixed integer non-convex optimization problem. The non-convexity has the following reasons. First, the optimization variables in the objective function and the constraints are coupled, e.g., C1 and C9. Second, the achievable rate for FBT in C1 has a non-convex structure. Finally, the integer constraints C6, C8, and C14 are non-convex. Such problems are in general NP hard and are known to be difficult to solve. However, in the next section, we propose an optimal scheme based on a branch-and-bound approach using monotonic optimization which finds the optimal solution of the considered problem. Moreover, in Section V, we propose two efficient suboptimal schemes that find excellent solutions and entail low computational complexity.

IV. PROPOSED GLOBAL OPTIMAL SOLUTION

In this section, we propose a branch-and-bound algorithm to solve problem (16) optimally. Different from the general branch-and-bound algorithms proposed for non-convex problems, e.g., [38], the proposed branch-and-bound algorithm exploits the monotonicity of the problem to reduce the search space for faster convergence [39]. The purpose of finding a global optimal solution to (16) is twofold: (1) determining a performance upper bound for OFDMA-URLLC MEC systems, and (2) having a benchmark for the efficient sub-optimal solutions presented in Section V. We first transform optimization problem (16) into the canonical form of discrete monotonic optimization. Then, we present the optimal algorithm based on a new branch-and-bound algorithm which aims to minimize an upper bound on the objective function of (16) until convergence to the optimal solution.

A. PROBLEM TRANSFORMATION

In this subsection, we transform problem (16) into the canonical form of a monotonic optimization problem. First, we introduce the following constraints in optimization problem (16):

\[
\begin{align*}
\text{C16: } & p^u_k[m^u, n^u] = s^u_k[m^u, n^u]p^u_k[m^u, n^u], \forall k, m^u, n^u, \\
\text{C17: } & p^d_k[m^d, n^d] = s^d_k[m^d, n^d]p^d_k[m^d, n^d], \forall k, m^d, n^d.
\end{align*}
\]

where both  \( k \) and  \( \alpha_k \) are non-negative. Constraints C1 and C2 guarantee the transmission of the required number of bits from user  \( k \) to the BS in the uplink and from the BS to user  \( k \) in the downlink, respectively, if the user offloads the task, i.e.,  \( \alpha_k = 0 \). Constraint C3 is the uplink-downlink causality constraint and constraint C4 ensures that user  \( k \) is served such that its task meets the associated delay requirement. Constraints C5 and C6 for the uplink and constraints C7 and C8 in the downlink are imposed to ensure that each sub-carrier in a given time slot is allocated to at most one user. Constraints C9 and C11 are the total transmit power constraints of user  \( k \) and the BS, respectively. Constraints C10 and C12 are the non-negative transmit power constraints. Constraint C13 ensures that the maximum allowed delay for local computing is not exceed when  \( \alpha_k = 1 \). Constraint C14 is the mode selection constraint. Finally, constraint C15 limits the CPU frequency of the local CPUs to  \( f_{\text{max}} \).

Remark 2: Resource allocation algorithm design for conventional MEC systems is typically based on Shannon’s capacity formula, i.e.,  \( V_c^k(s^u_k, p^u_k) \) and  \( V_d^k(s^d_k, p^d_k) \) in C1 and C2 are absent. The presence of  \( V_c^k(s^u_k, p^u_k) \) and  \( V_d^k(s^d_k, p^d_k) \) makes optimization problem (16) significantly more difficult to solve but is essential for capturing the characteristics of OFDMA-URLLC MEC systems.

Problem (16) is a mixed integer non-convex optimization problem. The non-convexity has the following reasons. First, the optimization variables in the objective function and the constraints are coupled, e.g., C1 and C9. Second, the achievable rate for FBT in C1 has a non-convex structure. Finally, the integer constraints C6, C8, and C14 are non-convex. Such problems are in general NP hard and are known to be difficult to solve. However, in the next section, we propose an optimal scheme based on a branch-and-bound approach using monotonic optimization which finds the optimal solution of the considered problem. Moreover, in Section V, we propose two efficient suboptimal schemes that find excellent solutions and entail low computational complexity.

Based on (17) and (18) optimization problem (16) is transformed into the following equivalent form:

\[
\begin{align*}
\text{minimize } & \sum_{k=1}^{K} w_k \left( \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} p^u_k[m^u, n^u] + (1 - \alpha_k) P_{\text{ex}} \right) + b_{\text{BS}} \sum_{k=1}^{K} \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} p^d_k[m^d, n^d] \\
\text{s.t. } & C1: F_c^k(p^u_k) - V_c^k(p^u_k) \geq (1 - \alpha_k) B_k, \forall k, \\
& C2: F_d^k(p^d_k) - V_d^k(p^d_k) \geq (1 - \alpha_k) V_k, \forall k, C3 - C8, \\
& C12 - C17, C9: \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} p^u_k[m^u, n^u] \leq P_{\text{max}}, \forall k, \\
& C10, C11: \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} p^d_k[m^d, n^d] \leq P_{\text{max}}.
\end{align*}
\]

where

\[
\begin{align*}
F_c^k(p^u_k) &= \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} \log_2 \left( 1 + \frac{\gamma_k}{m^u, n^u} \right), \\
V_c^k(p^u_k) &= \alpha Q^{-1} \left( \epsilon_k \right) \sum_{m^u=1}^{M^u} \sum_{n^u=1}^{N^u} v_c^k[m^u, n^u].
\end{align*}
\]

Optimization problem (19) is equivalent to (16) since the additional constraints (17) and (18) ensure that no power is allocated to a user over a sub-carrier if  \( s^u_k[m^u, n^u] = 0 \) or  \( s^d_k[m^d, n^d] = 0 \), respectively. Although optimization problem (19) is still non-convex, it is more tractable compared to equivalent problem (16), and as is shown in the following, it can be transformed into a monotonic optimization problem. To this end, we first study the monotonicity of problem (19) in the following two lemmas.

**Lemma 1:** Constraints C1 and C2 are differences of two monotonic and concave functions.

**Proof:** The proof closely follows a similar proof in [37], and is omitted here in the interest of space.

**Lemma 2 (See [47]):** Consider inequality  \( g(x) - h(x) \leq 0 \), where both \( g(x) \) and \( h(x) \) are increasing functions. Assuming  \( 0 \leq x \leq b \), then,  \( g(x) \leq g(b) \). Thus, there exist positive  \( t \) such that  \( g(x) + t \leq g(b) \). Therefore, inequality  \( g(x) - h(x) \leq 0 \) can
be split into two inequalities \( g(x) + t \leq g(b) \), \( h(x) + t \geq g(b) \), where \( 0 \leq t \leq g(b) \).

Exploiting Lemma 2, by defining positive auxiliary optimization variables \( 0 \leq \zeta^u \leq V_k^u(P_{k,\text{max}}), \forall k \), and \( 0 \leq \zeta^d \leq V_k^d(P_{k,\text{max}}), \forall k \), we transform non-monotonic constraints C1 and C2 into the following equivalent monotonic constraints:

\[
\begin{align*}
\text{C1a: } & F_k^u(p_k^u) + \zeta^u_k \geq V_k^u(P_{k,\text{max}}) + (1 - \alpha_k)B_k, \forall k,
\text{C1b: } & V_k^u(P_{k,\text{max}}), \forall k,
\text{C2a: } & F_k^d(p_k^d) + \zeta^d_k \geq V_k^d(P_{k,\text{max}}) + (1 - \alpha_k)Gamma_kB_k, \forall k,
\text{C2b: } & V_k^d(P_{k,\text{max}}), \forall k,
\end{align*}
\]

(22)

(23)

(24)

(25)

where \( V_k^u(P_{k,\text{max}}) \) is obtained by allocating all power available in the uplink, i.e., \( P_{k,\text{max}} \), to time slot \( n' \), sub-carrier \( m' \), and user \( k \). \( V_k^d(P_{k,\text{max}}) \) is defined in a similar way.

Now, optimization problem (19) can be transformed into the following equivalent form:

\[
\begin{align*}
\text{minimize } & \sum_{k=1}^{K} w_k \left( f_k^d + \delta_k \sum_{m'=1}^{M'} \sum_{n''=1}^{N''} p_k^{m'}[m'', n''] \right) + (1 - \alpha_k)P_{\text{cir}} + \delta_{\text{BS}} \sum_{k=1}^{K} \sum_{m'=1}^{M'} \sum_{n'=1}^{N'} p_k^{m'}[m', n'] \\
\text{s.t. } & \text{C1a, C1b, C2a, C2b, C3 - C17,}
\end{align*}
\]

(26)

where \( \zeta \) is the collection of optimization variables \( \zeta^u_k, \forall k, j \). In order to find an optimal solution for (26), we perform an exhaustive search over the binary variables in \( \alpha \). For a given \( \alpha_k = \tilde{\alpha}_k, \forall k \), optimization problem (26) reduces to the following optimization problem:

\[
\begin{align*}
\text{minimize } & \sum_{k=1}^{K} w_k \left( f_k^d + \delta_k \sum_{m'=1}^{M'} \sum_{n''=1}^{N''} \bar{p}_k^{m'}[m'', n''] \right) + (1 - \tilde{\alpha}_k)P_{\text{cir}} + \delta_{\text{BS}} \sum_{k=1}^{K} \sum_{m'=1}^{M'} \sum_{n'=1}^{N'} \bar{p}_k^{m'}[m', n'] \\
\text{s.t. } & \text{C1a, C1b, C2a, C2b, C3 - C13,}
\end{align*}
\]

(27)

The optimal solution of problem (26) can be obtained by solving problem (27) for all \( 2^K \) possible values of \( \alpha \). Then, we select that \( \alpha = \tilde{\alpha} \) which minimizes the objective function of (27). Problem (27) is in the canonical form of a discrete monotonic optimization problem. Moreover, to facilitate the design of an optimal algorithm for solving (27), we rewrite (27) in the following form:

\[
\begin{align*}
\text{minimize } & \tilde{\Phi} \\
\text{s.t. } & \mathcal{V} \in \mathcal{G} \cap \mathcal{H},
\end{align*}
\]

(28)

\( \tilde{\Phi} \) is the objective function in (27). Set \( \mathcal{G} \) is defined by constraints C1b, C2b, and C3 - C17, and co-normal set \( \mathcal{H} \) is defined by constraints C1a and C2a. The main difficulties in solving problem (28) are the reverse convex constraints C1b, C2b, and the non-convex binary constraints C6, C8. Moreover, for given \( (f, p^u, p^d, \zeta) \), problem (28) can be solved optimally in the remaining variables as we will explain in the following. Therefore, an efficient algorithm to find the optimal solution of (28) can be constructed by dividing optimization variables \( s^u, p^u, s^d, p^d, \zeta \) into two sets. The first set contains the convex variables \( f \) and \( \zeta \) and the non-convex variables \( p^u \) and \( p^d \) as the so-called outer variables, while the second set contains the binary variables \( s^u \) and \( s^d \) as the so-called inner variables. Furthermore, once \( p^u \) and \( p^d \) have been determined, according to (17), (18), we can obtain the values of \( s^u \) and \( s^d \) by comparing the values of the entries of \( p^u \) and \( p^d \) with zero. If the value of \( p_k^{m', n'} \) is greater than 0, this means that the corresponding \( s_k^{m', n'} = 1 \) if \( p_k^{m', n'} = 0 \). Moreover, for given \( f, p^u, p^d, \zeta \), problem (28) turns into the following feasibility check problem:

\[
\begin{align*}
\text{minimize } & 1 \\
\text{s.t. } & \mathcal{V} \in \mathcal{G} \cap \mathcal{H},
\end{align*}
\]

(29)

Since the values of \( s^u \) and \( s^d \) are known, we can simply check the constraint in (29).

B. DESIGN OF OPTIMAL ALGORITHM

Optimization problem (28) is a discrete monotonic optimization problem which can be optimally solved via the branch-and-bound algorithm as explained in the following [39], [48]. To facilitate the presentation of the optimal solution, we collect optimization variables \( (f, p^u, p^d, \zeta) \) in vector \( u \in \mathbb{R}^L \), where \( L = KM^uN^u + KM^dN^d + 2K \). The solution of (28) lies on the boundary of the feasible set, due to the monotonicity of the objective function and the constraints. However, the boundary of the feasible set is unknown. Thus, we approach the boundary by enclosing the feasible set \( \mathcal{V} = \mathcal{G} \cap \mathcal{H} \) by an initial box \( \mathcal{B}^{(0)} = [\mathcal{u}^{(0)}, \mathcal{u}^{(0)}] \), where \( \mathcal{u}^{(0)} \) and \( \mathcal{u}^{(0)} \) are lower and upper bounds, respectively, for the collection of variables \( u \). We ensure \( \mathcal{u}^{(0)} \) and \( \mathcal{u}^{(0)} \) to be contained in \( \mathcal{G} \setminus \mathcal{H} \) and \( \mathcal{H} \), respectively.

If this condition is not satisfied, either the problem is infeasible (when \( \mathcal{u}^{(0)} \) is not in set \( \mathcal{G} \)) or \( \mathcal{u}^{(0)} \) is an optimal solution of the problem (when \( \mathcal{u}^{(0)} \) is in \( \mathcal{V} \)). Iteratively, we split certain hyperrectangles, i.e., boxes, on the optimization variables \( u \) and try to improve a lower bound and an upper bound on the optimal value of the objective function. To aid this process, a local lower bound \( L_\mathcal{B} \) is stored for each box \( \mathcal{B} \in \mathcal{L} \), where \( \mathcal{L} \) is the set of all available boxes. Moreover, the current best value of the objective function obtained so far is denoted by \( C_\mathcal{B}^{\text{best}} \). An algorithmic description of the proposed branch-and-bound scheme is presented in Algorithm 1. In the following, we explain the algorithm in more detail.

1) Selection and Branching: In each iteration \( i \) of the optimal algorithm, i.e., in Line 3 of Algorithm 1, we start
Algorithm 1 Branch-and-Bound Algorithm

1: Initialization: Ensure $\bar{u}^{(0)} \in \mathcal{G} \setminus \mathcal{H}$ and $\bar{u}^{(0)} \in \mathcal{H}$. Set $B^{(0)} = \{\bar{u}^{(0)}, \bar{u}^{(0)}\}$, $\mathcal{L} = \{B^{(0)}\}$, $L_B(B^{(0)}) = \Phi(\bar{u}^{(0)})$, $C_{BV}(B^{(0)}) = \tilde{\Phi}(\bar{u}^{(0)})$, $\mathcal{S}$ denotes a set of feasible solutions, and maximum iteration number $I_{\text{max}}$. 

2: for $i = 1 : I_{\text{max}}$ do 

3: Selection and branching: Select box $B^{(i)} = [\bar{u}^{(i)}, \bar{u}^{(i)}] \in \mathcal{L}$ such that $B^{(i)} = \arg\min_{B \in \mathcal{L}} \Phi(\bar{u})$ and branch it into two new boxes $B_{1}^{(i)}$ and $B_{2}^{(i)}$. 

4: Feasibility check of the two new boxes: 

5: for $b = 1 : 2$ do 

6: suppose $B_{b}^{(i)} = [\bar{u}_{b}^{(i)}, \bar{u}_{b}^{(i)}]$ 

7: calculate local lower bound $L_{B_{b},b}^{(i)}$ for $B_{b}^{(i)}$ 

8: if $(L_{B_{b},b}^{(i)} < C_{BV})$ then 

9: check the feasibility of lower corner $\bar{u}_{b}^{(i)}$ by solving (29) 

10: if lower corner $\bar{u}_{b}^{(i)}$ is feasible 

11: update $C_{BV} = L_{B_{b},b}^{(i)}$ and store the feasible solution $\bar{u}_{b}^{(i)}$, i.e., $\mathcal{S} \leftarrow \bar{u}_{b}^{(i)}$, 

12: else 

13: if $\bar{u}_{b}^{(i)} \in \mathcal{G}$ and $\bar{u}_{b}^{(i)} \notin \mathcal{H}$ 

14: the box may be feasible, i.e., may contain feasible solutions 

15: else 

16: the box is not feasible and cannot contain any feasible solution 

17: end if 

18: end if 

19: end if 

20: end for 

21: Bounding and Pruning: Update the set of boxes $\mathcal{L}$ for the next iteration of the algorithm 

22: for each $B \in \mathcal{L}$ do 

23: if $L_{B_{b},b}^{(i)} > C_{BV}$ 

24: Remove $B_{b}^{(i)}$ 

25: end if 

26: remove the branched box ($\mathcal{L} \leftarrow \mathcal{L} \setminus B^{(i)}$) and remove infeasible boxes 

27: end for 

28: $i \leftarrow i + 1$ 

29: end for 

30: Output: Optimal solution $u^* = \arg\min_{\mathcal{S}} \tilde{\Phi}(u)$. 

by selecting the box $B^{(i)}$ that has the lowest lower bound from the set of available boxes $\mathcal{L}$ as follows:

$$B^{(i)} = \arg\min_{B \in \mathcal{L}} \tilde{\Phi}(\bar{u}).$$

(30)

After selecting a box $B^{(i)} = [\bar{u}^{(i)}, \bar{u}^{(i)}]$, we bisect the longest edge of $B^{(i)}$. We first calculate

$$\tilde{j} = \arg\max_{j=1,\ldots,L} \left\{ \left\| \bar{u}^{(i)} - \bar{u}^{(j)} \right\| \right\},$$

(31)

then, $B^{(i)}$ is partitioned into two new boxes as follows [49]:

$$B_1^{(i)} = \left[ u^{(i)}, \bar{u}^{(i)} - \frac{\bar{u}^{(i)} - \bar{u}^{(j)}}{2} \right] e_j,$$

(32)

$$B_2^{(i)} = \left[ u^{(i)} + \frac{\bar{u}^{(i)} - \bar{u}^{(j)}}{2} \right] e_j, \bar{u}^{(i)} \right\],$$

(33)

where $e_j \in \mathbb{R}^L$ is a vector whose $j$-th element is equal to one and the remaining elements are zero. The bisection rule in (32) guarantees that the branching process is exhaustive [39, 49, 50] and the algorithm converges to the optimal solution.

2) Feasibility Check: After the two new boxes $B_1^{(i)} = [u^{(i)}, \bar{u}^{(i)}]$ and $B_2^{(i)} = [\bar{u}^{(i)}, \bar{u}^{(i)}]$ are generated, we check the lower and upper corners of each box and verify whether these boxes are feasible or not, see Lines 4-20. To do so, we first calculate local lower bounds $L_{B_{b},b}^{(i)} = \tilde{\Phi}(\bar{u}_{b}^{(i)}), \forall b \in \{1, 2\}$ for $B_{1}^{(i)}$ and $B_{2}^{(i)}$, respectively, see Line 7. Subsequently, we compare the values of the local lower bounds $L_{B_{b},b}^{(i)}, \forall b \in \{1, 2\}$ with the best global value $C_{BV}$ obtained so far. If the local lower bound of one of the two new boxes is greater than $C_{BV}$, then this box can be removed. On the other hand, if the local lower bound is smaller than $C_{BV}$, we check the feasibility of the box and search for better feasible points. To do so, we first check the lower corners of each box by checking the feasibility of (29). If the lower corners are feasible, then, these lower corners will be added to the set of feasible solutions $\mathcal{S}$ and we update the current best value $C_{BV}$. Otherwise, if this condition is not satisfied, we check if the box contains feasible solutions. The box is not feasible if $u^{(i)} \notin \mathcal{G}$ or $\bar{u}^{(i)} \notin \mathcal{H}$. In this case, we remove the infeasible box in the next step of the algorithm, i.e., in the pruning step.

Remark 3: Although variables $\zeta$ and $f$ are convex variables, we branch over them. In fact, this facilitates the optimal algorithm design and reduces the total computation time needed for finding the optimal solution as it eliminates the use of convex software solvers which would contribute significantly to the overall computation time.

3) Bounding and Pruning: The bounding and pruning steps are described in the following:

Bounding: The problem is to find lower and upper bounds for $\Phi(\bar{u})$ over the set $\mathcal{G} \cap \mathcal{H}$ for a given box $B = [\bar{u}, \bar{u}]$. Due to the monotonicity of $\Phi(\cdot)$, we can obtain the upper and lower bounds as $\Phi(\bar{u})$ and $\tilde{\Phi}(\bar{u})$, respectively.

Pruning: In the pruning step infeasible boxes are removed. These boxes have local lower bounds greater than the current best global value, i.e., $L_{B_{b},b}^{(i)} > C_{BV}, \forall b$, and the original branched box in iteration $i$, i.e., $B^{(i)}$. This step is performed to reduce memory consumption and to achieve faster convergence.

C. CONVERGENCE COMPLEXITY ANALYSIS

Convergence Analysis: For a sufficiently large number of iterations $I_{\text{max}}$, Algorithm 1 is guaranteed to find a globally
optimal solution of (28). This can be shown following the same steps as the convergence analysis of the monotonic branch-and-bound algorithms in [47], [51]. Specifically, we first recall that the branching and reduction operations have the same properties as those in [47], [51]. In particular, they guarantee that the upper and lower bounds of \( \Phi \) in each box are always improved in every iteration (branching rule), and that no feasible point in a box is lost (reduction operation). On the other hand, for the bounding step, the feasibility is checked based on (29), the calculation of the tighter upper bound is always improved in every iteration (branching rule), and the upper bound is non-increasing. Therefore, following the proof in [47], Algorithm 1 generates a sequence of boxes such that the gap between the upper bound and lower bound is guaranteed to converge to a single point, which is a globally optimal solution of (28).

**Complexity Analysis:** The computational complexity of Algorithm 1 is exponential in the number of variables of the optimization problem. Thus, the complexity order of Algorithm 1 is \( O(2^L) \). Due to its high complexity, the proposed optimal resource allocation algorithm cannot be used in real time applications, especially for URLLC systems. However, it provides a valuable performance benchmark for low-complexity suboptimal algorithms. In the next section, we focus on developing low-complexity resource allocation algorithms based on SCA to strike a balance between computational complexity and performance.

V. SCA-BASED SUBOPTIMAL SOLUTIONS

In this section, we propose two low-complexity suboptimal algorithms based on SCA.

**A. PROPOSED SCA-BASED SUBOPTIMAL SCHEME 1**

In this sub-section, we propose a suboptimal algorithm that tackles the non-convexity of (16) in three main steps. First, we use the Big-M formulation to linearize the product terms \( s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d], \forall k, m^d, n^d, \forall j \). Then, we employ difference of convex (DC) programming and SCA methods to find a locally optimal solution of optimization problem (16).

1) **Big-M Formulation:** The product \( s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d], \forall k, m^d, n^d, \forall j \), in (16) is non-convex. Fortunately, the product of a binary variable \( s \) and a continuous variable \( p \) can be transformed to linear inequality constraints using the Big-M formulation [52, Sec. 2.3]. To this end, we first need to define a new variable \( \bar{p} = sp \). Then, using the Big-M formulation [52, Sec. 2.3], the non-convex product is rewritten in terms of equivalent linear inequality constraints as follows: \( \bar{p} \leq p_{\text{max}}, \bar{p} \leq p, \bar{p} \geq p - (1 - s)p_{\text{max}}, \) and \( \bar{p} \geq 0, \) where \( p_{\text{max}} \) is an upper bound for variable \( p \). To tackle the non-convexity of \( s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d], \forall j, \) based on the above considerations, we define new optimization variables

\[
\bar{p}_k^G[m^d, n^d] = s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d], \forall k, m^d, n^d, \forall j. \tag{34}
\]

Now, we decompose the product terms in (34) using the Big-M formulation and impose the following additional constraints [52]:

\[
\begin{align*}
C16 : & \quad \bar{p}_k^G[m^u, n^u] \leq \bar{p}_k^G[m^u, n^u], \forall k, m^d, n^d, \tag{35} \\
C17 : & \quad \bar{p}_k^G[m^u, n^u] \leq \bar{p}_k^G[m^u, n^u], \forall k, m^u, n^u, \tag{36} \\
C18 : & \quad \bar{p}_k^G[m^u, n^u] \geq \bar{p}_k^G[m^u, n^u] - (1 - s_k^{\prime}[m^d, n^d])P_{\text{max}}, \forall k, m^d, n^d, \tag{37} \\
C19 : & \quad \bar{p}_k^G[m^d, n^d] \geq 0, \forall k, m^d, n^d, \tag{38} \\
C20 : & \quad \bar{p}_k^G[m^d, n^d] \leq P_{\text{max}}s_k^{\prime}[m^d, n^d], \forall k, m^d, n^d, \tag{39} \\
C21 : & \quad \bar{p}_k^G[m^d, n^d] \leq \bar{p}_k^G[m^d, n^d], \forall k, m^d, n^d, \tag{40} \\
C22 : & \quad \bar{p}_k^G[m^d, n^d] \geq 0, \forall k, m^d, n^d, \tag{41} \\
C23 : & \quad \bar{p}_k^G[m^d, n^d] \geq \bar{p}_k^G[m^d, n^d] - (1 - s_k^{\prime}[m^d, n^d])P_{\text{max}}, \forall k, m^d, n^d. \tag{42}
\end{align*}
\]

In this manner, the non-convex product term \( s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d], \forall k, m^d, n^d, \forall j \) in (34) is transformed into a set of linear inequalities. Note that constraints C16-C23 do not change the feasible set. Now, optimization problem (16) is transformed into the following equivalent form:

\[
\begin{align*}
\text{minimize} & \quad \hat{\Phi} \\
\text{s.t.} & \quad \tilde{C}1: \quad \tilde{F}_1^G(\tilde{p}_k^G) - \tilde{V}_1^G(\tilde{p}_k^G) \geq (1 - \alpha_k)B_k, \forall k, \\
& \quad \tilde{C}2: \quad \tilde{F}_2^G(\tilde{p}_k^G) - \tilde{V}_2^G(\tilde{p}_k^G) \geq (1 - \alpha_k)\Gamma_kB_k, \forall k, \forall C = 8, \\
& \quad \tilde{C}9: \quad \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} \bar{p}_k^G[m^d, n^d] \leq P_{\text{max}}, \forall k, \\
& \quad \tilde{C}10, \tilde{C}11: \quad \sum_{k=1}^{K} \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} \bar{p}_k^G[m^d, n^d] \leq \tilde{P}_{\text{max}}, \forall k, \\
& \quad \tilde{C}12 - \tilde{C}23, \tag{43}
\end{align*}
\]

where

\[
\hat{\Phi} = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{j=1}^{J} \bar{w}_k^G[m^d, n^d]s_k^{\prime}[m^d, n^d]p_k^G[m^d, n^d] + (1 - \alpha_k)P_{\text{car}} \delta_{B_k} + \delta_{\text{BS}} \sum_{k=1}^{K} \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} \bar{p}_k^G[m^d, n^d]. \tag{44}
\]

\[
\tilde{F}_1^G(\tilde{p}_k^G) = \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} \log_2(1 + \tilde{\gamma}_k^G[m^d, n^d]), \tag{45}
\]

\[
\tilde{V}_1^G(\tilde{p}_k^G) = \alpha Q^{-1}(\epsilon_k) \sum_{m^d=1}^{M^d} \sum_{n^d=1}^{N^d} \tilde{\gamma}_k^G[m^d, n^d]. \tag{46}
\]
2) DC Programming: The two remaining difficulties in solving problem (43) are the binary variables in constraints C6, C8, and C14 and the structure of the achievable FBT rate in C1 and C2. To tackle these issues, we employ a DC programming approach [37], [53], [54], [55]. To this end, the integer constraints in (43) are rewritten in terms of DC functions as follows:

\begin{align}
C6a : & \quad 0 \leq s_k^\ell [m^\ell, n^\ell] \leq 1, \forall k, m^\ell, n^\ell, \tag{47} \\
C6b : & \quad E^\ell(s^\ell) - H^\ell(s^\ell) \leq 0, \tag{48} \\
C8a : & \quad 0 \leq s_k^\ell [m^\ell, n^\ell] \leq 1, \forall k, m^\ell, n^\ell, \tag{49} \\
C8b : & \quad E^\ell(s^\ell) - H^\ell(s^\ell) \leq 0, \tag{50} \\
C14a : & \quad 0 \leq \alpha_k \leq 1, \forall k, \tag{51} \\
C14b : & \quad E^\alpha(\alpha) - H^\alpha(\alpha) \leq 0, \tag{52}
\end{align}

where

\begin{align}
E^\ell(s^\ell) = & \sum_{k=1}^K \sum_{m^\ell=1}^{M^\ell} \sum_{n^\ell=1}^{N^\ell} s^\ell_k [m^\ell, n^\ell], \forall j, \tag{53} \\
H^\ell(s^\ell) = & \sum_{k=1}^K \sum_{m^\ell=1}^{M^\ell} \sum_{n^\ell=1}^{N^\ell} \left( s^\ell_k [m^\ell, n^\ell] \right)^2, \forall j, \tag{54} \\
\text{and} \\
E^\alpha(\alpha) = & \sum_{k=1}^K \alpha^2_k, \quad H^\alpha(\alpha) = \sum_{k=1}^K \alpha_k. \tag{55}
\end{align}

Now, constraints C6, C8, and C14 are equivalently formulated in continuous form, cf. C6a, C8a, and C14a. However, constraints C6b, C8b, and C14b are still non-convex, as they are reverse convex constraints. In order to handle them, we introduce the following lemma.

Lemma 3: For sufficiently large constant values \( \eta_1, \eta_2 \), and \( \eta_3 \), problem (43) is equivalent to the following problem:

\begin{align*}
\text{minimize} & \quad \Phi + \beta\left(s^\ell, s^d, \alpha\right) \\
\text{s.t.} & \quad C1, \ C2, C3 - C5, C6a, C7, C8a, C9, \ C10 - C15, C14a, C17 - C23, \tag{56}
\end{align*}

where

\begin{align}
\beta\left(s^\ell, s^d, \alpha\right) = & \eta_1\left(E^\ell(s^\ell) - H^\ell(s^\ell)\right) \\
& + \eta_2\left(E^d(s^d) - H^d(s^d)\right) \\
& + \eta_3\left(E^\alpha(\alpha) - H^\alpha(\alpha)\right). \tag{57}
\end{align}

Proof: Please refer to Appendix. 

Constants \( \eta_1, \eta_2, \) and \( \eta_3 \) act as penalty factors to penalize the objective function for any \( s^\ell_k [m^\ell, n^\ell] \) and \( \alpha_k \) that is not equal to 0 or 1. The remaining sources of non-convexity are the structure of the achievable FBT rate, and the non-convex objective function. In the following, we employ SCA to approximate problem (56) by a convex problem. Subsequently, we propose an iterative algorithm to find a low-complexity solution.

3) SCA: In order to tackle the remaining non-convexity of (56), we employ the Taylor series approximation to approximate the non-convex parts of the objective function and constraints \( C1 \) and \( C2 \). Since \( H^\ell(s^\ell), \forall j, -V^1_j(\Phi_k), \forall j, \) and \( H^\alpha(\alpha) \) are differentiable convex functions, then for any feasible points \( s^\ell(0), \Phi_k^0, \forall j, \) and \( \alpha(0) \), where the superscript \( i \) denotes the SCA iteration index, the following inequalities hold:

\begin{align}
H^\ell(s^\ell) & \geq \tilde{H}^\ell(s^\ell, s^\ell(0)) = H^\ell(s^\ell(0)) + \nabla s^\ell H^\ell(s^\ell(0))^T (s^\ell - s^\ell(0)), \forall j, \tag{58} \\
V^j_k(\Phi_k) & \leq \tilde{V}^j_k(\Phi_k^0) = V^j_k(\Phi_k^0) + \nabla \Phi_k V^j_k(\Phi_k^0)^T (\Phi_k - \Phi_k^0), \forall j. \tag{59}
\end{align}

and

\begin{align}
H^\alpha(\alpha) & \geq \tilde{H}^\alpha(\alpha, \alpha(0)) = H(\alpha(0)) + \nabla \alpha H(\alpha(0))^T (\alpha - \alpha(0)). \tag{60}
\end{align}

The right hand sides of (58), (59), and (60) are affine functions representing the global underestimation of \( H^\ell(s^\ell), \forall j, \), \( V^j_k(\Phi_k), \) and \( H^\alpha(\alpha) \), respectively, where \( \nabla \Phi_k V^j_k(\Phi_k^0) \) are the gradients of \( H^\ell(s^\ell) \) and \( V^j_k(\Phi_k^0) \), respectively. By substituting the right hand sides of (58)-(60) into (56), we obtain the following optimization problem:

\begin{align*}
\text{minimize} & \quad \Phi + \tilde{\beta}(s^\ell, s^\ell(0), s^d, s^d(0), \alpha, \alpha(0)) \\
\text{s.t.} & \quad C1, \ C2, C3 - C5, C6a, C7, C8a, C9, \ C10 - C15, C14a, C17 - C23, \tag{61}
\end{align*}

where \( \tilde{\beta}(s^\ell, s^\ell(0), s^d, s^d(0), \alpha, \alpha(0)) = \eta_1(E^\ell(s^\ell) - \tilde{H}^\ell(s^\ell, s^\ell(0)) + \eta_2(E^d(s^d) - \tilde{H}^d(s^d, s^d(0))) + \eta_3(E^\alpha(\alpha) - \tilde{H}^\alpha(\alpha, \alpha(0))). \) Optimization problem (61) is a convex optimization problem. To facilitate the application of CVX for solving problem (61), we reformulate the cubic function \( f^3_k \) appearing in the cost function and transform it into two equivalent SOC constraints [58]. We first define new auxiliary variables \( \zeta_k, \forall k, \) to upper bound the cubic function as follows \( f^3_k \leq \zeta_k, \forall k. \) Then, as shown in [58], we can expand \( f^3_k \leq \zeta_k, \forall k, \) to the following equivalent SOC constraints [58]:

\begin{align}
C24 : & \quad \left[ \begin{array}{c} \zeta_k \\ \hat{\theta}_k \end{array} \right] \geq 0, \forall k, \\
C25 : & \quad \left[ \begin{array}{c} \hat{\theta}_k \\ f_k \\ 1 \end{array} \right] \geq 0, \forall k, \tag{62}
\end{align}

where \( \hat{\theta}_k, \forall k, \) are new auxiliary variables. Optimization problem (61) is transformed into the following equivalent form:

\begin{align*}
\text{minimize} & \quad \Phi + \tilde{\beta}(s^\ell, s^\ell(0), s^d, s^d(0), \alpha, \alpha(0)) \\
\text{s.t.} & \quad C1, \ C2, C3 - C25, \tag{63}
\end{align*}
Algorithm 2: Successive Convex Approximation

1: Initialize: Random initial points \( s_0^{(1)}, s_1^{(1)}, \bar{p}_1^{(1)}, \bar{p}_2^{(1)}, \alpha^{(1)} \). Set iteration index \( i = 1 \), maximum number of iterations \( \ell_{\text{max}} \), and penalty factors \( \eta_1 > 0, \eta_2 > 0 \), and \( \eta_3 > 0 \).

2: Repeat

3: Solve convex problem (63) for given \( s_i^{(i)}, s_i^{(i)}, \bar{p}_i^{(i)}, \bar{p}_i^{(i)}, \alpha^{(i)} \), and store the intermediate solutions \( s_i = s_i^{(i)}, s_i^d = s_i^{(i)}, \bar{p}_i = \bar{p}_i^{(i)}, \bar{p}_i^d = \bar{p}_i^{(i)}, \alpha^* = \alpha^{(i)} \).

4: Set \( i = i + 1 \) and update \( s_i = s_i^{(i)}, s_i^d = s_i^{(i)}, \bar{p}_i = \bar{p}_i^{(i)}, \bar{p}_i^d = \bar{p}_i^{(i)}, \alpha^* = \alpha^{(i)} \).

5: Until \( i = \ell_{\text{max}} \).

6: Output: \( s_*, s_\delta^* = s_i^d, \bar{p}_\delta^* = \bar{p}_i^d, \alpha^* = \alpha^{(i)} \).

where

\[
\Phi = \sum_{k=1}^{K} w_k \bar{\zeta}_k + \frac{\varepsilon_0}{\bar{\nu}} \left( \frac{1}{\alpha_k^{(i)}} \right) \left( 1 - \frac{1}{\left( 1 + \gamma[i] \right)^2} \right) \approx 1,
\]

is accurate when the received SNR \( \gamma[i] \), exceeds 5 dB as is typically the case in cellular networks, especially when supporting URLLC [61], [62], [63]. On the other hand, in the low SNR regime, by substituting \( v[i] = \bar{\nu}[i] = 1 \) in (3), we obtain a lower bound on the achievable rate. If the lower bound is used for optimization of the resource allocation in MEC systems, the feasibility of the obtained solution is guaranteed. Hence, exploiting this approximation, we rewrite the expression for the dispersion for uplink and downlink transmission in optimization problem (16) as follows:

\[
\tilde{V}_k^*(s_k^*) = a Q^{-1}(\epsilon_k^*) \sum_{m=1}^{M_k} \sum_{n=1}^{N_k} s_k[m_k^*, n_k^*], \forall j = \{u, d\}.
\]

Now, defining \( \bar{p}_j^*[m_k^*, n_k^*] = s_k^*[m_k^*, n_k^*] p_j^*[m_k^*, n_k^*], \forall k, m_k^*, n_k^*, \forall j \in \{u, d\} \), as new optimization variables, and rewriting \( \tilde{V}_k^*(s_k^*) \) in (66) as \( \tilde{V}_k^*(s_k^*) \), optimization problem (16) can be transformed as follows:

\[
\begin{align*}
\text{minimize} & \quad \Phi \\
\text{s.t.} & \quad \tilde{C}_1: \tilde{F}_j^*(s_k^*, \bar{p}_k^*) - \tilde{V}_k^*(s_k^*) \geq (1 - \alpha_k) B_k, \forall k, \\
& \quad \tilde{C}_2: \tilde{F}_j^*(s_k^*, \bar{p}_k^*) - \tilde{V}_k^*(s_k^*) \geq (1 - \alpha_k) C_k, \forall k, \\
& \quad C_3 - C_8, C_9: \sum_{m=1}^{M_k} \sum_{n=1}^{N_k} \bar{p}_j^*[m_k^*, n_k^*] \leq P_{k, \text{max}}, \forall k, \\
& \quad \tilde{C}_{10}: \tilde{p}_j^*[m_k^*, n_k^*] \geq 0, \forall k, m_k^*, n_k^*, \\
& \quad \tilde{C}_{11}: \tilde{p}_j^*[m_k^*, n_k^*] \geq 0, \forall k, m_k^*, n_k^*, \text{C13 - C15},
\end{align*}
\]

B. PROPOSED SCA-BASED SUBOPTIMAL SCHEME 2

For suboptimal scheme 1, we have adopted the Big-M method to linearize non-convex product terms. However, this method introduced additional optimization variables and constraints, which negatively affect the complexity of Algorithm 2. In this subsection, we reduce the complexity of suboptimal scheme 1 (Algorithm 2). To do so, we first approximate the dispersion in the high SNR regime as follows:

\[
\bar{\nu}[i] = \left( 1 - \frac{1}{\left( 1 + \gamma[i] \right)^2} \right) \approx 1,
\]

and \( \bar{\zeta} \) and \( \bar{\theta} \) are the collections of optimization variables \( \bar{\zeta}_k \), \( \bar{\theta}_k \), respectively. Optimization problem (63) is convex because the objective function is convex and can be efficiently solved by standard convex optimization solvers such as CVX [58]. Algorithm 2 summarizes the main steps for solving (65) in an iterative manner, where the solution of (63) in iteration \( i \) is used as the initial point for the next iteration \( i + 1 \). By iteratively solving (63), Algorithm 2 produces a sequence of improved feasible solutions, which for sufficiently large \( \ell_{\text{max}} \) converge to a local optimum point of problem (56) or equivalent problem (16) in polynomial time [59], [60].
In optimization problem (67), we use similar techniques as constraints C6, C8, and C14 and the cubic function present C1b and C2b in suboptimal scheme 1. As a consequence, optimization is convex because the objective function is convex and the constraints span a convex set since constraints C1b and C2b can be represented as SOCs. To deal with constraints C6, C8, and C14 and the cubic function present in optimization problem (67), we use similar techniques as in suboptimal scheme 1. As a consequence, optimization problem (67) can be rewritten in the following equivalent form:

$$\text{minimize} \quad \tilde{\Phi} + \tilde{\beta}(s^u, s^{d(i)}, s^d, s^{d(i)}, \alpha, \alpha^i)$$

s.t. C1a, C1b, C2a, C2b, C3 - C5,

C6a: $s_k^u [m^u, n^u] \in [0, 1], \forall k, m^u, n^u, C7,$

C8a: $s_k^d [m^d, n^d] \in [0, 1], \forall k, m^d, n^d,$

C9 - C13,

C14a: $\alpha_k \in [0, 1], \forall k,$

C15, C16: $\tilde{\xi}_k, \tilde{\eta}_k, \tilde{\theta}_k \geq 0, \forall k,$

C17: $\frac{\tilde{\xi}_k}{\tilde{\eta}_k, \tilde{\theta}_k} \geq 0, \forall k,$

where

$$\tilde{\Phi} = \sum_{k=1}^{K} w_k \left( \kappa \xi_k + \delta_k \sum_{i=1}^{N^u} \sum_{m^u=1}^{M^u} \tilde{p}_k^u[m^u, n^u] + (1 - \alpha_k)P_{cw} \right) + \delta_{BS} \sum_{k=1}^{K} \sum_{i=1}^{M^d} \sum_{n^d=1}^{N^d} \tilde{p}_k^d[m^d, n^d].$$

and $z$ and $q$ are the collection of optimization variables $z_k$, $\forall k$, and $q_k$, $\forall k$, respectively. Optimization problem (74) is convex because the objective function is convex and the constraints span a convex set. Therefore, it can be efficiently solved by standard convex optimization solvers such as CVX [58]. Algorithm 3 summarizes the main steps for solving (67) in an iterative manner, where the solution of (74) in iteration (i) is used as the initial point for the next iteration (i + 1). The algorithm produces a sequence of improved feasible solutions until convergence to a local optimum point of problem (67). Unlike Algorithm 2, Algorithm 3 does not provide a local optimum solution to problem (16) because of the approximation of the dispersion term. Nevertheless, Algorithm 3 provides an upper bound on the total system power consumption and the obtained solution is feasible for (16). Moreover, this upper bound becomes tight for sufficiently high SNR, where the approximation in (65) becomes tight, which is likely the case for URLLC applications.

C. COMPLEXITY ANALYSIS OF SUBOPTIMAL ALGORITHMS

In this sub-section, we study the complexity of the proposed low-complexity suboptimal schemes.

1) Algorithm 1: Optimization problem (63) is a non-linear convex problem which can be solved efficiently in polynomial time using, e.g., CVX [58]. There are in total $3KM^uN^u + 3KM^dN^d + 4K$ optimization variables and $M^uN^u(2 + 7K) + M^dN^d(2 + 7K) + 8K + KM^uM^d\bar{O}$ linear and convex constraints. Thus, the computational complexity order of Algorithm 2 per iteration is $O((3KM^uN^u + 3KM^dN^d + 4K)(M^uN^u(M^uN^u(2 + 7K) + M^dN^d(2 + 7K) + 8K + KM^uM^d\bar{O})))$ [64], [65], [66].

2) Algorithm 3: Optimization problem (74) is also a non-linear convex problem, which can be solved efficiently in polynomial time using, e.g., CVX [58]. There are in total $2KM^uN^u + 2KM^dN^d + 6K$ optimization variables and $M^uN^u(2 + 2K) + M^dN^d(2 + 2K) + 8K + KM^uM^d\bar{O}$ linear and convex constraints. Thus, the computational complexity order of Algorithm 3 per iteration is $O((2KM^uN^u + 2KM^dN^d + 6K)(M^uN^u(2 + 2K) + M^dN^d(2 + 2K) + 8K + KM^uM^d\bar{O})))$ [64], [65], [66]. As can be observed, the complexity of Algorithm 3 is lower than that of Algorithm 2. The reason behind this is the smaller number of optimization variables and constraints in (74) compared to (63).

Remark 4: Algorithm 1 is mainly used as a benchmark for low-complexity solutions such as Algorithms 2 and 3. Algorithm 2 provides a better performance but has a higher complexity than Algorithm 3. Hence, in a practical application, depending on the relative importance of complexity and performance, Algorithm 2 or Algorithm 3 may be preferred.

VI. PERFORMANCE EVALUATION

In this section, we provide simulation results to evaluate the performance of the proposed joint uplink-downlink resource allocation algorithm for OFDMA-URLLC MEC systems. We adopt the simulation parameters provided in Table 1, unless specified otherwise. In our simulations, a single cell is considered with inner and outer radii of $r_1$ and $r_2$, respectively.
TABLE 1. Default values of simulation parameters.

| Parameter                           | Value          | Parameter                           | Value          |
|-------------------------------------|----------------|-------------------------------------|----------------|
| Total number of sub-carriers in uplink and downlink $M = M^u = M^d$ | $M = 2, M^u = 64$ | Number of time slots in uplink and downlink $N^u = N^d$ | 4             |
| Bandwidth of each sub-carrier       | 36 kHz         | Noise power density                 | -174 dBm/Hz    |
| Maximum BS transmit power $P_{\text{max}}$ | 45 dBm         | Maximum transmit power of each user $P_{\text{max}}$ | 25 dBm         |
| Packet error probability            | $c_{l_j} = 10^{-3}, \forall j, k$ | Circuit power consumption of user $k$, $P_{\text{cu}}$ | 50 mW [45]     |
| Value of $\nu_j, \forall j$        | 1              | Effective switched capacitance $\kappa$ | $10^{-27}$ Farad |
| Required CPU cycles for processing one bit of information $c_k$ | $[100 - 5000]$ cycles/bit [43] | Maximum CPU frequency of local user processor $f_{\text{cpu}}$ | 2.7 GHz        |
| Power amplifier inefficiency of the users and the BS $\alpha = 1, \forall k$, and $b_{\text{BS}} = 1$ | Users weights $w_k = 1, \forall k$ |

The BS is located at the center of the cell, and the URLLC users are randomly located between the inner and the outer radii. The path loss is calculated as $35.3 + 37.6 \log_{10}(d_k)$ [63], where $d_k$ is the distance from the BS to user $k$. The values of the penalty factors used in Algorithms 2 and 3 are set to $\eta_1 = 10P_{e_{\text{max}}}$ and $\eta_2 = \eta_3 = 10P_{e_{\text{max}}}$. The small scale fading gains between the BS and the URLLC users are modelled as independent and identically Rayleigh distributed. All simulation results are averaged over 1000 realizations of the path loss and multipath fading.

A. PERFORMANCE BOUND AND BENCHMARK SCHEMES

We compare the performance of the proposed resource allocation algorithms with the following schemes:

- **Shannon’s capacity (SC) [22]**: To obtain an (unachievable) lower bound on the total network power consumption, Shannon’s capacity formula is adopted in problem (16), i.e., $V_j^H(k, p_k^j), \forall j$, is set to zero in constraints C1 and C2, respectively, and all other constraints are retained. The resulting optimization problem is solved using a modified version of Algorithm 2.

- **Local computation (LC)**: In this scheme, only local computation is employed where each user attempts to minimize its local computation power by optimizing its own CPU frequency subject to its delay constraint. The resulting optimization problem is convex and can be solved optimally using convex optimization tools such as CVX [58].

- **Edge Only (EO)**: In this scheme, all URLLC users offload their data to the edge server. The resulting optimization problem is solved using the SCA based algorithm from [1].

- **Fixed Sub-Carrier Assignment (FSA)**: In this scheme, we fix the sub-carrier assignment for offloading and optimize the remaining degrees of freedom via SCA. We divide the total number of sub-carriers among the users such that their delay and causality constraints are met. This can be done by solving a mixed integer feasibility problem.

B. CONVERGENCE OF THE PROPOSED ALGORITHMS

In Figs. 2 and 3, we investigate the convergence of the proposed optimal algorithm (Algorithm 1) and the suboptimal algorithms (Algorithms 2 and 3) for different numbers of sub-carriers $M^u, M^d$, and different numbers of users $K$ for a given channel realization. We show the total sum power consumption as a function of the number of iterations. As can be observed from Fig. 2, the proposed optimal scheme converges to the global optimal solution after a finite number of iterations. In particular, the optimal scheme converges after 100000 and 170000 iterations for $M_T = 24$ and $M_T = 32$, respectively. For the proposed optimal scheme, the number
of iterations required for convergence increases significantly with the number of sub-carriers since increasing the number of sub-carriers increases the dimensionality of the search space. On the other hand, the proposed suboptimal scheme 1 (Algorithm 2) attains an excellent performance for a much smaller number of iterations. We note that optimization problem (27) has to be solved $2^K$ times to find the global optimal solution, see Section IV-B. We show in Fig. 2 the solution for the best $\hat{\alpha}$.

In Fig. 2, we chose relatively small values for $M^l, \forall j, N^l, \forall j,$ and $K$ since the complexity of the optimal algorithm increases rapidly with the dimensionality of the problem. In Fig. 3, we investigate the convergence behavior of the proposed suboptimal schemes for larger parameters values. As can be observed from Fig. 3, for all considered parameter values, the proposed suboptimal schemes require a small number of iterations to converge. In particular, the proposed suboptimal scheme 1 requires at most 4 iterations to converge while the proposed suboptimal scheme 2 requires only 2 iterations. The reason for the faster convergence of the suboptimal scheme 2 is the convexity of the feasible set of the underlying optimization problem (74), while for suboptimal scheme 1, the feasible set of the corresponding optimization problem (63) is an approximated convex set, and thus, the algorithm requires more iteration to converge. On the other hand, suboptimal scheme 2 causes a higher power consumption compared to suboptimal scheme 1. The higher power consumption is caused by the approximation of channel dispersion in (65) used for derivation of suboptimal scheme 2 which yields an upper bound on the achievable power consumption. As expected, the convergence speeds of the proposed suboptimal schemes are less sensitive to the problem size and the number of users compared to that of the optimal scheme as they avoid the costly branching operation of branch-and-bound type algorithms.

**C. AVERAGE SYSTEM POWER CONSUMPTION VERSUS TASK SIZE**

In Figs. 4 and 5, we investigate the average system power consumption versus the task size of the URLLC users. As expected, increasing the required number of computed bits leads to a higher power consumption. This is due to the fact that if more bits are to be transmitted or computed in a given frame, higher SNRs or high CPU frequencies are needed, and thus, the BS and the users consume more power.

In Fig. 4, we compare the performance of the proposed schemes with SC. SC provides a lower bound for the required power consumption of OFDMA-URLLC MEC systems. However, SC cannot guarantee the required latency and reliability. This is due to the fact that, in this scheme, the performance loss incurred by FBT is not taken into account for resource allocation design, and thus the obtained resource allocation policies may not meet the QoS constraints. As can be seen, the proposed suboptimal schemes attain an excellent performance. Thereby, suboptimal scheme 1 achieves a lower average system power consumption than suboptimal scheme 2 since the latter approximates the dispersion as in (65). On the other hand, as pointed out in Section V-C, suboptimal scheme 2 entails a low computational complexity. Hence, the proposed suboptimal schemes offer different trade-offs between performance and complexity.

In Fig. 4, we chose relatively small values for $K, M^l, M^d, N^l, N^d, and B_k, \forall k$, since the complexity of the optimal Algorithm 1 increases rapidly with the dimensionality of the problem, cf. Section IV-D. In Fig. 5, we investigate the performance of the proposed suboptimal schemes for larger parameters values. As can be seen, the proposed schemes lead to a substantially lower power consumption compared to the FSA, LC, and EO baseline. For the FSA scheme, the poor performance is due to the smaller number of degrees of freedom for resource allocation as this scheme uses a fixed sub-carrier allocation. For the LC scheme, the
performance degradation is caused by the limited computation capability of the URLLC users’ CPUs. The proposed schemes also attain large power savings compared to the EO scheme. This is due to the joint optimization of local and edge computing, while for the EO scheme only offloading is considered.

D. IMPACT OF CELL RADIUS

In Fig. 6, we study the impact of the outer cell radius on the average system power consumption for different resource allocation schemes. As can be observed, increasing the outer cell radius increases the average system power consumption. This is due to the fact that the path loss increases with the distance, and as a result, more power is needed to maintain the same SNR for larger distances. For small outer radii, the performance of the proposed scheme is close to that of the EO scheme, as in this case, the proposed scheme is likely to offload the tasks of the users to the edge server because of the low transmission power needed. However, as the outer cell radius increases, the path loss increases, and thus the users are more likely to compute their tasks locally to reduce power consumption. In this case, the performance of the proposed scheme approaches that of the LC scheme. In Fig. 6 also shows the impact of \( \Gamma \) on the offloading probability. As can be seen, the offloading probability decreases with \( \Gamma \), which is due to the fact that as \( \Gamma \) increases, the size of the computed results in the downlink becomes larger, and the BS has to allocate more power to satisfy the QoS constraint in the downlink. In this case, the users are more likely to compute their tasks locally in order to limit the total system power consumption which leads to a lower offloading probability.

E. IMPACT OF DELAY AND VALUE OF OFFSET \( \tau \)

In Fig. 8, we investigate the impact of different delay requirements and consider three delay scenarios. For delay scenario \( S_0 \), all users have the same delay requirements, i.e., \( D_k = 6, \forall k \). For delay scenario \( S_1 \), we have \( D_k = 6, \forall k = \{2, 3, 4\} \). For delay scenario \( S_2 \), we have \( D_k = 6, \forall k = \{1, 2, 3\} \), and \( D_4 = 6 \). In Fig. 8, we show the average system power consumption versus delay parameter \( D \). As can be observed, the average system power consumption decreases with \( D \), which is due to the fact that a larger \( D \) increases the feasible set of problem (16) and increases the flexibility of resource allocation. Moreover, the proposed suboptimal scheme attains large power savings compared to LC, especially, when the users have strict delay requirement. This is due to the limited computation capability of the users.

In Fig. 9, we investigate the impact of \( \tau \) on the average system power consumption for different resource allocation schemes and \( N^u = N^d = 4 \). As can be seen, the average system power consumption decreases as the value of \( \tau \) increases. The reason for this behaviour is that the number of overlapping time slots \( \bar{O} = N^u - \tau \) is reduced as \( \tau \) increases.
and the feasible set of optimization problem (16) become larger at the expense of an increase in the latency of the users, $D_k = \tau + n^d$. On the other hand, for small values of $\tau$, the number of overlapping time slots increases, and the total system latency is reduced for all users. This causes the average system power consumption to increase.

VII. CONCLUSION AND FUTURE WORK

In this paper, we studied the resource allocation algorithm design for OFDMA-URLLC MEC systems. To ensure the stringent end-to-end transmission delay and reliability requirements of URLLC MEC systems, we proposed a joint uplink-downlink resource allocation scheme that takes FBT into account. Moreover, to minimize the end-to-end delay, we proposed a partial time overlap between the uplink and downlink frames which introduces a new uplink-downlink causality constraint. The proposed resource allocation algorithm design was formulated as an optimization problem for minimization of the total weighted transmit power of the network under QoS constraints regarding the minimum required number of computed bits of the URLLC users within a maximum computation time. The resulting optimization problem was shown to be a non-convex mixed-integer problem and hard to solve. Nevertheless, we solved the optimization problem optimally using a branch-and-bound technique based on monotonic optimization theory. Moreover, to strike a balance between computation complexity and performance, we proposed two efficient suboptimal low-complexity schemes based on SCA. Our simulation results showed that the proposed resource allocation algorithm design facilitates the application of URLLC in MEC systems, and achieves significant power savings compared to several benchmark schemes. Moreover, our simulation results showed that the proposed suboptimal algorithms offer different trade-offs between performance and complexity and attained an excellent performance at comparatively low complexity.

In this paper, we provide optimal and low-complexity sub-optimal algorithms for resource allocation in OFDMA-URLLC MEC systems. An interesting topic for future work is the investigation of machine learning techniques to enable fast online resource allocation in such systems to avoid having to solve the considered optimization problem for each channel realization [67]. In this context, the algorithms proposed in this paper can be exploited to obtain training data for supervised deep learning algorithms [68]. In addition, other machine learning techniques, such as reinforcement learning algorithms [28], [29] or a combination of supervised and unsupervised deep learning techniques [69], may be exploited for resource allocation in OFDMA-URLLC MEC systems.

APPENDIX

The proof of Lemma 3 follows similar steps as corresponding proofs in [37],[53],[54]. In the following, we show that problems (43) and (56) are equivalent. Let $U^*$ denote the optimal objective value of (56). We define the Lagrangian function of problem (43), denoted by $\mathcal{L}(f, \bar{p}^u, \bar{p}^d, s^u, s^d, \alpha, \eta_1, \eta_2, \eta_3)$, as [70]:

$$
\mathcal{L}(f, \bar{p}^u, \bar{p}^d, s^u, s^d, \alpha, \eta_1, \eta_2, \eta_3) = \Phi \left( f, \bar{p}^u, \bar{p}^d, \alpha \right) + \eta_1 \left( E^u(s^u) - H^u(s^u) \right) + \eta_2 \left( E^d(s^d) - H^d(s^d) \right) + \eta_3 \left( E^a(\alpha) - H^a(\alpha) \right),
$$

(76)

where $\eta_1, \eta_2$, and $\eta_3$ are the Lagrange multipliers corresponding to constraints C6b, C8b and C14b, respectively. Note that $E^u(s^u) - H^u(s^u) \geq 0$, $E^d(s^d) - H^d(s^d) \geq 0$, and $E^a(\alpha) - H^a(\alpha) \geq 0$ hold. Using Lagrange duality [37],[55],[70], we have the following relation:

$$
U^* = \max_{\eta_1, \eta_2, \eta_3 \geq 0} \min_{\bar{p}^u, \bar{p}^d, s^u, s^d, \alpha} \mathcal{L}(f, \bar{p}^u, \bar{p}^d, s^u, s^d, \alpha, \eta_1, \eta_2, \eta_3)
$$

(77)

7. Note that weak duality holds for convex and non-convex optimization problems [70].
\[
\min_{\mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3} \max_{\alpha, \eta_1, \eta_2, \eta_3} \mathcal{L}(\mathbf{f}, \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3) = U^*,
\]

(78)

where \( \Omega \) is the feasible set specified by the constraints in (43). In the following, we first prove the strong duality, i.e., \( U_d = U^* \). Let \((\mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{p}^\alpha, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3)\) denote the solution of (77). For this solution, the two cases are possible. \textit{Case 1)} If \( E^d(s^u) - H^d(s^u) > 0 \), \( E^d(s^d) - H^d(s^d) > 0 \), and \( E^\alpha(\alpha) - H^\alpha(\alpha) > 0 \) hold, the optimal \( \eta_1^*, \eta_2^*, \) and \( \eta_3^* \) are infinite, respectively. Hence, \( U_d \) is infinite too, which contradicts the fact that it is upper bounded by a finite-value \( U^* \). \textit{Case 2)} If \( E^d(s^u) - H^d(s^u) = 0 \), \( E^d(s^d) - H^d(s^d) = 0 \), and \( E^\alpha(\alpha) - H^\alpha(\alpha) = 0 \) holds, then \((\mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{p}^\alpha, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3)\) belongs to the feasible set of the original problem (43) which implies \( U_d = U^* \). Hence, strong duality holds, and we can focus on solving the dual problem (77) instead of the primal problem (78).

Next, we show that any \( \eta_1 \geq \eta_1^0, \eta_2 \geq \eta_2^0, \) and \( \eta_3 \geq \eta_3^0 \) are optimal solutions of dual problem (77), i.e., \( \eta_1^*, \eta_2^*, \) and \( \eta_3^* \), where \( \eta_1^0, \eta_2^0, \) and \( \eta_3^0 \) are some sufficiently large numbers. To do so, we show that \( \Theta(\eta_1, \eta_2, \eta_3) \triangleq \mathcal{L}(\mathbf{f}, \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3) \) is a monotonically increasing function of \( \eta_1, \eta_2, \) and \( \eta_3 \). Recall that \( E^d(s^u) - H^d(s^u) \geq 0 \), \( E^d(s^d) - H^d(s^d) \geq 0 \), and \( E^\alpha(\alpha) - H^\alpha(\alpha) \geq 0 \) holds for any given \( \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3 \). Therefore, \( \mathcal{L}(\mathbf{f}, \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3) \leq \mathcal{L}(\mathbf{f}, \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1(1), \eta_2(1), \eta_3(1)) \leq \Theta(\eta_1(1), \eta_2(1), \eta_3(1)) \) holds for any \( \mathbf{p}^u, \mathbf{p}^s, \mathbf{s}^u, \mathbf{s}^d, \mathbf{f}, \alpha, \eta_1, \eta_2, \eta_3 \in \Omega \). Therefore, due to strong duality, we can use the dual problem (56) to find the solution of the primal problem (43) and any \( \eta_1 \geq \eta_1^0, \eta_2 \geq \eta_2^0, \) and \( \eta_3 \geq \eta_3^0 \) are optimal dual variables. These results are concisely given in Lemma 3 which concludes the proof.

REFERENCES

[1] W. Ghanem, V. Jamali, Q. Zhang, and R. Schober, “Joint uplink–downlink resource allocation for OFDMA-URLLC MEC systems,” in Proc IEEE Global Commun. Conf., Taipei, Taiwan, Dec. 2020, pp. 1–7.

[2] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” IEEE Netw., vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[3] P. Popovski, “Ultra-reliable communication in 5G wireless systems,” in Proc IEEE Int. Conf. 5G Ubiquitous Connect, Nov. 2014, pp. 146–151.

[4] Y. Polyanskiy, “Channel coding: Non-asymptotic fundamental limits,” Ph.D. dissertation, Dept. Elect. Eng., Princeton Univ., Princeton, NJ, USA, 2010.

[5] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel coding rate in the finite blocklength regime,” IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307–2359, May 2010.

[6] W. R. Ghanem, V. Jamali, Y. Sun, and R. Schober, “Resource allocation for multi-user downlink MISO OFDMA-URLLC systems,” IEEE Trans. Commun., vol. 68, no. 11, pp. 7184–7200, Nov. 2020.
