Université du Québec
Institut national de la recherche scientifique
Centre Énergie, Matériaux et Télécommunications

RESOURCE MANAGEMENT FOR ENABLING HETEROGENEOUS SERVICES AND APPLICATIONS IN WIRELESS CELLULAR SYSTEMS

Par

NGUYEN TI TI

Thèse présentée pour l'obtention du grade de
Doctorat en philosophie, Ph.D.
en télécommunications

Jury d'évaluation

Examinateur externe	Prof. Nghi H. Tran *University of Akron*
	Prof. Hien Quoc Ngo *Queen's University Belfast*
Examinateur interne	Prof. André Girard *INRS-ÉMT*
Directeur de recherche	Prof. Long Bao Le *INRS-ÉMT*

Acknowledgments

I would like to gratefully acknowledge and express a sincere thanks to my supervisor, Professor Long Bao Le for giving me the opportunity to pursue doctoral study at INRS-ÉMT, University of Québec. I am truly fortunate to benefit from his remarkable technical knowledge and passion for science. From the very first day, he has always pointed me in good research directions and motivated me to pursue them to concrete results. His invaluable support and guidance during my study have definitely helped me complete this Ph.D. dissertation. I would like to express my gratitude to other members of my Ph.D. committee – Professor André Girard of INRS-ÉMT, University of Québec who has regularly reviewed and constructively commented on the progress of my doctoral study. I would also like to thank Professor Nghi H. Tran of University of Akron and Professor Hien Quoc Ngo of Queen's University Belfast for serving as the external examiner to my Ph.D. dissertation.

I would like to express gratitude to all my colleagues for the magnificent and unforgettable time at the Networks and Cyber Physical Systems Lab (NECPHY-Lab), INRS-ÉMT, University of Québec: Vu Ha, Tuong Hoang, Hieu Nguyen, Dai Nguyen, Tam Tran, Thinh Tran, Tri Nguyen, Dat Nguyen, Hoang Vu, Tung Phan, and Thong Vo. Additionally, many thanks to my friends for helping me with the French translation to my Ph.D. dissertation.

Lastly, my sincere love and gratitude are devoted to all of my family members: Mom and Dad, my brothers and sisters, who always support me in all my life endeavours. Without my family's constant and unconditional support, my Ph.D. study would not be completed. I thank you all and hope that I made you proud of my accomplishments.

Abstract

5G New Radio (NR) and Mobile Edge Computing (MEC) have been recently proposed as important technologies and architectures for the next-generation wireless cellular system, which allows efficiently supporting emerging services and applications. Indeed, 5G NR provides a flexible frame structure with scalable Transmission Time Interval (TTI) in different so-called numerologies, which enable us to meet diverse quality of service (QoS) requirements of different wireless services and compute-intensive applications. However, many challenges must be resolved to efficiently utilize different types of system resources and better support heterogeneous wireless services. The overall objective of this Ph.D. research is to develop efficient resource management techniques for next-generation systems exploiting 5G NR and MEC technologies. Our research has resulted in three major research contributions, which are presented in three corresponding main chapters of this dissertation.

First, we study fair computation offloading and resource allocation for the MIMO based MEC system, which is presented in Chapter 5. In particular, we formulate the joint computation offloading and resource allocation problem that minimizes the maximum weighted consumed energy for mobile users considering the latency and resource limitation constraints. Then, we propose different efficient algorithms to solve the underlying mixed-integer nonlinear programming (MINLP) problems under perfect and imperfect channel state information estimations.

Second, we investigate the joint data compression, computation offloading, and resource allocation problem for hierarchical fog-cloud systems aiming to minimize the maximum weighted energy and service delay cost of all users, which is covered in Chapter 6. To this end, we propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by data compression and decompression. Then, we first consider the scenario where data compression is performed only at the mobile users. A novel three-step approach using convexification techniques is developed to optimize the compression ratios and the resource allocation. As the next step, we address the more general design where data compression is performed at both the mobile users and the fog server. We propose three algorithms to overcome the strong coupling between the offloading decision and the resource allocation and find efficient solutions for the underlying problem..

Finally, we consider leveraging the 5G NR to support diverse applications with different requirements. The research outcomes of this study are presented in Chapter 7. In particular, 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. Then, two algorithms, namely Resource Partitioning-based

Algorithm (RPA) and Iterative Greedy Algorithm (IGA), are developed to acquire efficient resource scheduling solutions.

For all the considered problems and proposed designs, extensive numerical results are presented to gain further insights and to evaluate the performance of our algorithms. Our numerical studies confirm that our algorithms can achieve efficient resource utilization, energy saving, and significant performance gains compared to existing designs.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Extended Summary

1.1 Background and Motivation

The research on the fifth-generation (5G) and beyond wireless cellular networks has been driven by the need to support the mobile traffic explosion and the rapidly increasing number of wireless communications, including both human-based and Internet-of-Things (IoT) connections. Specifically, it is predicted that tens of billions of wireless devices, from low-cost IoT to smartphones, tablets, virtual reality headsets, and cars will be connected to wireless networks over the next few years [2]. The communication demand on different kinds of mobile devices in vertical domains including the smart factory, smart vehicles, smart grid, smart city is more and more sophisticated. Thus, future wireless networks must provide different communication services with different QoS requirements. In particular, International Telecommunication Union (ITU) classifies 5G mobile network services into three categories: enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC), and ultra-Reliable and Low Latency Communications (uRLLC). In general, eMBB refers to bandwidth-hungry services, such as high-definition video and virtual/augmented reality (VR/AR) streamings. As such, mMTC is suitable in scenarios with dense connectivity, such as smart cities and smart farming. This is in distinction to uRLLC, which is aimed at supporting the time-sensitive networking and mission-critical services, such as automatic/assisted driving and remote control. To provide these services, a new frame structure has been defined in 5G New Radio (NR). By allowing the flexible numerology, the Transmission Time Interval (TTI) and frame-size can be flexibly configured to fit with the service demands [3]. Some early papers [4–6] on the

scheduling and resource allocation with 5G NR frame structure show a promising way to enhance the future system performance.

In general, mobile edge/cloud computing (MEC/MCC) technologies enable enhancing the mobile usability and prolonging the mobile battery life by offloading computation-intensive applications to a remote fog/cloud server [7–9]. In an MCC system, enormous computing resources are available in the core network, but the limited backhaul capacity can induce significant delay for the underlying applications. In contrast, an MEC system, with computing resources being deployed at the network edge in close proximity to the mobile devices, can enable computation offloading and meet demanding application requirements [10]. Hierarchical fog-cloud computing systems which leverage the advantages of both MCC and MEC can further enhance the system performance [11–15] where fog servers deployed at the network edge can operate collaboratively with the more powerful cloud servers to execute computation-intensive user applications. Specifically, when the users' applications require high computing power or low latency, their computation tasks can be offloaded and processed at the fog and/or remote cloud servers. The potential scenarios and applications of MEC and its variants are still being discussed for 5G and beyond systems.

Due to the need for data exchange incurred in the offloading process, the wireless transmission plays an integral role in the computational offloading system [16]. Therefore, to efficiently utilize MEC power, one needs to develop efficient designs for joint management of both wireless and computing resources. Moreover, advanced communications technologies such as massive multiple-input multiple-output (MIMO), heterogeneous network (HetNet), and device-to-device (D2D), which allow enhancing the spectral efficiency will be employed in MEC and its variants. Accordingly, the joint management of two different kinds of resources becomes very challenging. Specifically, different from the general wireless networks, the energy and time delay in MEC and its variants are related not only to wireless transmission but also to computation factors. They are complicated functions of different parameters and factors such as bandwidth, transmit power, central processing unit (CPU) clock speed, execution location. Effective management and optimization of these parameters in two different kinds of resources is a very challenging problem[17], and still requires significantly more concerted effort from the wireless community [18, 19].

Besides, an important aspect of 5G wireless network is the application's view. Indeed, with recent breakthroughs in artificial intelligence (AI), new emerging applications enabling new ways of

interactions among things and humans have been created to enhance the quality of life. Many of them are compute-intensive applications such as e-health, object recognition/detection/monitoring. When only communications-related issues are concerned in network design and management, it is impossible to enable these compute-intensive applications on many different kinds of devices, especially low-cost IoT devices. Therefore, 5G wireless networks must support not only communication, but also computation, control, and content delivery (4C) functions. Mobile Edge Computing (MEC) has been recently proposed as an important technology in 5G wireless networks to enable a variety of new compute-intensive applications even on low-cost IoT devices. In general, MEC is a network architecture concept defined by the european telecommunications standards institute (ETSI) [20], that enables cloud computing capabilities and an information technology (IT) service environment at the edge of the cellular network. Different design aspects of MEC, such as task partitioning and resource allocation, have been investigated in both academic and industry communities to enable them and support future system scenarios and applications [21, 22].

5G NR is an entirely new air interface being developed for 5G to support a wide variety of services and devices. This part will briefly introduce a new concept in 5G NR, named numerology. Note that numerology is a term which is used to define the grid of discrete resources in the continuous time-frequency plane. A critical feature in 5G NR is the utilization of carrier frequency from sub-1 GHz up to 52.6 GHz, as defined in the 3rd Generation Partnership Project (3GPP) Release 15. However, the channel propagation properties of low and high frequency bands are very different. In particular, the low frequency band is strongly affected by the delay spread intensive environments while the high frequency band is strongly influenced by the phase noise [23]. Accordingly, a single numerology applied for a wide range of frequencies becomes inefficient or even impossible. Comparing to LTE numerology, 5G NR supports multiple types of subcarrier spacing based on a baseline subcarrier spacing of 15 kHz [24]. In particular, 5G NR defines five distinct OFDM numerologies which is parameterized as μ, $\mu = 0, 1, 2, 3, 4$. The numerology μ has the subcarrier bandwidth of $2^{\mu} \times 15$ kHz and the slot duration of $2^{-\mu}$ milliseconds. In numerology $\mu = 0$, the time-frequency grid is the same with LTE, 5G NR can be coexistence with LTE and the LTE-based NB-IoT on the same subcarrier. For the lower frequency bands with narrow subcarrier spacing, numerologies 0, 1, and 2 are used to counter the delay spread intensive environments. For the higher frequency bands, using numerology 2, 3, and 4 with wide subcarrier spacing can make the system robust to the phase noise, and can support low latency services efficiently [24]. The introduction of different

numerologies provides the flexibility for various services in the same system. It also introduces new challenges when multiplexing different numerologies in the same time-frequency space. Due to the different time-frequency grid, the wireless scheduling for heterogeneous services with mixed numerology in the 5G wireless networks becomes an important problem to improve the system performance.

1.2 Research Contributions

Computation and radio resource management for MEC systems and wireless scheduling for heterogeneous services considering 5G NR mixed numerology are research problems considered in the dissertation. To this end, we consider three fundamental design aspects to allow the above coexistence, namely resource allocation and offloading decision in MEC systems, joint data compression, offloading decision, and resource allocation in hierarchial fog-cloud systems, and wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks. Specifically, our main contributions can be described as follows.

1.2.1 Computation Offloading in MIMO Based MEC Systems Under Perfect and Imperfect CSI Estimation

In this contribution, we develop algorithms for optimizing the offloading and resource allocation in MIMO based MEC systems considering perfect/imperfect-CSI estimation. Existing computation offloading designs for the multi-task multi-user scenario have not considered the important MIMO communication technology and its related issues such as the imperfect CSI estimation. Our current paper aims to fill this gap in the literature by proposing general offloading and resource allocation algorithms which can provide fairness and consider the cutting-edge MIMO technology. In particular, the main contributions of our work can be summarized as follows:

- We consider two important scenarios with perfect-CSI (P-CSI) and imperfect-CSI (IP-CSI) estimation for the MIMO-based MEC system.
- We propose different efficient algorithms to solve the joint computation offloading and resource allocation problem that minimizes the maximum weighted consumed energy (Min-max

W.C.E) for mobile users considering the latency and resource limitation constraints. In particular, we propose an optimal algorithm achieving the global optimal solution for P-CSI scenario, and two iterative low-complexity algorithms to determine the sub-optimal solutions for P-CSI and IP-CSI scenarios, respectively.

- We discuss the extension of the proposed design when the time to send back the computation outcomes of offloaded users are considered. We also describe how to extend the proposed algorithm to address this more general problem.

1.2.1.1 System Model

We consider an MEC system comprising one base station (BS), co-located with the edge server, equipped with M antennas and K single-antenna user equipments (UEs). It is noted that the design can be extended for UEs equipped with multiple antennas when the small scale fading is independent between all antennas and all users. We assume that UE k has the set of $\mathcal{L}_k$ independent computation tasks, and each task $l_k \in \mathcal{L}_k$ needs a number of CPU cycles c_{k,l_k} and a number of data bits b_{k,l_k} (to transmit the programming states to the BS). The binary offloading decision for task $l_k \in \mathcal{L}_k$ is captured by binary variable s_{k,l_k}, where $s_{k,l_k} = 1$ if task l_k is executed at the mobile device, and $s_{k,l_k} = 0$ if this task is offloaded to the edge server. Let f_k and f_k^{c} (CPU cycles/seconds) denote the CPU clock speed to execute the application of UE k locally in mobile device and remotely in edge server, respectively. Then, for user k, we have the local computation energy $\xi_k^{\mathsf{lo}} = \alpha_k f_k^2 \sum\limits_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}$ (Joules), the local execution time $t_k^{\mathsf{lo}} = f_k^{-1} \sum\limits_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}$ (seconds), and the remote execution time $t_k^{\mathsf{c}} = (f_k^{\mathsf{c}})^{-1} \sum\limits_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) c_{k,l_k}$ (seconds), where α_k denotes the energy coefficient specified in the CPU model.

Let $\boldsymbol{h}_k \in \mathbb{C}^{M \times 1}$ denote the uplink channel gain vector between UE k and the BS's antennas where elements of the uplink channel gain vector $\boldsymbol{h}_k$ are modeled as $h_{mk} = \varphi_{mk}\sqrt{\beta_k}, m \in \{1, 2, ..., M\}$, where φ_{mk} and β_k represent the small-scale and large-scale fading coefficients, respectively. Let p_k denote the uplink transmit power of UE k and $\boldsymbol{n}$ denote the noise vector whose components are i.i.d. $\mathcal{CN}(0, \sigma_{bs})$ variables. Let the set of UEs that cannot execute all their tasks locally be denoted as $\mathcal{K}_\xi$=$\{k \in \mathcal{K} | \sum_{l \in \mathcal{L}_k} s_{k,l_k} < |\mathcal{L}_k|\}$ and ZF based detection is applied in this considered system, the lower bound of the average transmission rate, the upper bound of the average transmission time

and the average transmission energy due to UE k for P-CSI scenario are respectively denoted as

$$r_k^{\mathsf{lb}} = W\log_2(1+p_k\beta_k^{\mathsf{a}}), \tag{1.1}$$

$$t_{k,\mathsf{P}}^{\mathsf{t,ub}} = (r_k^{\mathsf{lb}})^{-1}\sum_{l_k\in\mathcal{L}_k}(1-s_{k,l_k})b_{k,l_k}, \tag{1.2}$$

$$\xi_{k,\mathsf{P}}^{\mathsf{t,ub}} = (p_k+p_{k,0})t_{k,\mathsf{P}}^{\mathsf{t,ub}}, \tag{1.3}$$

where $\beta_k^{\mathsf{a}} = \frac{(M-|\mathcal{K}_\xi|)\beta_k}{\sigma_{bs}}$, W is the transmission bandwidth, p_k and $p_{k,0}$ are the transmit and circuit powers of user k, respectively. It is noted that the lower bound of the average transmission rate is determined based on the convexity of function $\log_2(1+1/x)$ and Jensen's inequality.

For IP-CSI scenario, let T denote the number of symbol periods corresponding to the channel coherence interval and let τ denote the number of symbols in the pilot. Let $\sqrt{\tau p^{\mathsf{tr}}}\boldsymbol{\phi}_k \in \mathbb{C}^{\tau\times 1}$ be the pilot sequence assigned for UE k where p^{tr} denotes the pilot power and $||\boldsymbol{\phi}_k||^2 = 1$. Assuming that ZF based detection is applied[1], $\boldsymbol{\phi}_k^H\boldsymbol{\phi}_j = 0, \forall k\neq j$ and $\tau \geq K_\xi$, the lower bound of average rate $\hat{r}_k^{\mathsf{lb}}$ is given as [25]:

$$\hat{r}_k^{\mathsf{lb}} = W\log_2\left(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k+p_k\right) - W\log_2\left(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k\right), \tag{1.4}$$

where $\lambda_{k,i} = \frac{(\tau p^{\mathsf{tr}}\beta_k+\sigma_{bs})\sigma_{bs}\beta_i}{(\tau p^{\mathsf{tr}}\beta_i+\sigma_{bs})\tau p^{\mathsf{tr}}\beta_k^2(M-K_\xi)}$, $\sigma_k = \frac{(\tau p^{\mathsf{tr}}\beta_k+\sigma_{bs})\sigma_{bs}}{\tau p^{\mathsf{tr}}\beta_k^2(M-K_\xi)}$. Then, the upper bounds on the average transmission time and energy (including both training and data transmission) can be written, respectively, as

$$t_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \frac{T}{T-\tau}t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{1.5}$$

$$\xi_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \left(\frac{\tau(p^{\mathsf{tr}}+p_{k,0})}{T-\tau}+p_k+p_{k,0}\right)t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{1.6}$$

where $t_{k,\mathsf{IP1}}^{\mathsf{t,ub}} = (\hat{r}_k^{\mathsf{lb}})^{-1}\sum_{l_k\in\mathcal{L}_k}(1-s_{k,l_k})b_{k,l_k}$.

The problem which minimizes the maximum weighted energy consumption (min-max W.C.E) at mobile users considering latency and limited computation-radio resource constraints can be stated

[1]Zero-forcing (ZF) decoder is one of three well-known methods in MIMO to mitigate the user interference, besides the maximum ratio transmission/maximum ratio combining (MRT/MRC) and the minimum mean-squared error (MMSE) [25]. It is studied heavily for IEEE 802.11n (MIMO). ZF will be useful when ISI is significant compared to noise. In case of imperfect CSI, the structure of the tranmission rates achieved by using ZF and MRC are similar. Therefore, our work can be applied for MRC. Besides, the performance of ZF and MMSE are quite similar when the number of antennas is large enough in [25]. Therefore, understanding ZF thoroughly is really useful and can give more insight information to system performance analysis.

as follows:

$$
\begin{aligned}
(\mathcal{P}_2) \quad & \min_{\boldsymbol{S},\boldsymbol{f},\boldsymbol{f}^{\mathsf{c}},\boldsymbol{p},\xi} \quad \xi && \text{(1.7a)}\\
& \text{s. t.} \quad w_k(\xi_k^{\mathsf{lo}} + \xi_k^{\mathsf{t}}) \leq \xi, \ \forall k, && \text{(1.7b)}\\
& \qquad\quad t_k^{\mathsf{lo}} \leq \eta_k, \quad \forall k, && \text{(1.7c)}\\
& \qquad\quad t_k^{\mathsf{t}} + t_k^{\mathsf{c}} \leq \eta_k, \ \forall k, && \text{(1.7d)}\\
& \qquad\quad s_{k,l_k} \in \{0,1\}, \quad \forall k, && \text{(1.7e)}\\
& \qquad\quad \textstyle\sum_{k\in\mathcal{K}_\xi} f_k^{\mathsf{c}} \leq F^{\mathsf{c}}, \quad f_k^{\mathsf{c}} \geq 0, && \text{(1.7f)}\\
& \qquad\quad 0 \leq f_k \leq F_k^{\mathsf{max}}, \quad \forall k, , && \text{(1.7g)}\\
& \qquad\quad 0 \leq p_k \leq p_k^{\mathsf{max}}, \quad \forall k, && \text{(1.7h)}
\end{aligned}
$$

where ξ be the min-max W.C.E, w_k denotes the energy weight of UE k, $\boldsymbol{S} = \{\boldsymbol{s}_k, \forall k\}, \boldsymbol{s}_k = \{s_{k,l_k}, \forall l_k\}$, $\{\boldsymbol{f}, \boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}\} = \{f_k, f_k^{\mathsf{c}}, p_k, \forall k\}$, η_k is the maximum allowable delay of UE k, F_k^{max} denotes the maximum computation capacity of UE k, and p_k^{max} represents the maximum transmit power of UE k, F^{c} is the available computing budget at edge server, t_k^{t} and ξ_k^{t} can be expressed for the P-CSI and IP-CSI scenarios as

$$
t_k^{\mathsf{t}} = \begin{cases} t_{k,\mathsf{P}}^{\mathsf{t,ub}}, \text{ P-CSI} \\ t_{k,\mathsf{IP}}^{\mathsf{t,ub}}, \text{ IP-CSI} \end{cases} \quad ; \quad \xi_k^{\mathsf{t}} = \begin{cases} \xi_{k,\mathsf{P}}^{\mathsf{t,ub}}, \text{ P-CSI} \\ \xi_{k,\mathsf{IP}}^{\mathsf{t,ub}}, \text{ IP-CSI} \end{cases} .
$$

Proposition 1.1. *Problem* $(\mathcal{P}_2)$ *can be recast as*

$$
\begin{aligned}
(\mathcal{P}_2) \quad & \min_{\boldsymbol{S},\boldsymbol{f}^{\mathsf{c}},\boldsymbol{p},\xi} \quad \xi \\
& \quad s.\ t. \quad \sum_{l_k\in\mathcal{L}_k} s_{k,l_k} c_{k,l_k} \leq \eta_k F_k^{\mathsf{max}}, \quad \forall k \in \mathcal{K}, && \text{(1.8a)}\\
& \qquad\quad (1.7b), (1.7d) - (1.7f), (1.7h),
\end{aligned}
$$

where

$$
\xi_k^{\mathsf{lo}} = \alpha_k \eta_k^{-2} \left(\sum_{l_k\in\mathcal{L}_k} s_{k,l_k} c_{k,l_k} \right)^3, \quad \forall k \in \mathcal{K}, \tag{1.9}
$$

Algorithm 1.1. Optimal Algorithm - P-CSI (P-O)

1: **Initialize**: choose ϵ, $\xi_{\min}=0$ and $\xi_{\max}=\min(\max(w_k\xi_k^{\mathsf{lo}}|s_{k,l_k}=1,\sum_{l_k\in\mathcal{L}_k}c_{k,l_k}\leq\eta_k F_k^{\mathsf{max}}),\xi^{\infty})$.

2: **while** $\xi_{\max}-\xi_{\min}<\epsilon$ **do**
3: Assign $\xi=(\xi_{\max}+\xi_{\min})/2$.
4: Determine set $\mathcal{K}_\xi$ as in (1.10).
5: Solve $(\mathcal{P}_3)_k$ to get $f_{k,\xi}^{\mathsf{c,min}}$ for all $k\in\mathcal{K}_\xi$.
6: Assign $feasibility=true$ if all subproblems $(\mathcal{P}_3)_k$ are feasible and $\sum_{k\in\mathcal{K}_\xi}f_{k,\xi}^{\mathsf{c,min}}\leq F^{\mathsf{c}}$.
7: Assign $(\xi_{\max},\xi_{\min})=\text{bisectionSearch}(feasibility,\xi)$
8: **end while**

1.2.1.2 Algorithm Design for P-CSI Scenario

To solve the difficult MINLP $(\mathcal{P}_2)$, we propose two algorithms where the first one (P-O) can find the global optimal solution while the second one (P-SO) achieves a solution with lower complexity.

a) P-CSI - Optimal Algorithm (P-O) :

Proposition 1.2. *For a given value of ξ, problem $(\mathcal{P}_2)$ is feasible if all subproblems $(\mathcal{P}_3)_k,\forall k$ are feasible and $\sum\limits_{k\in\mathcal{K}_\xi}f_{k,\xi}^{\mathsf{c,min}}\leq F^{\mathsf{c}}$ where $f_{k,\xi}^{\mathsf{c,min}}$ is the optimal value of $(\mathcal{P}_3)_k$, $\mathcal{K}_\xi$ is the the set of UEs who upload their computation tasks if the local execution consumes the total energy greater than ξ, which is computed as follows:*

$$|\mathcal{K}_\xi|=\sum_k\delta_k,\ \textit{and}\ \delta_k=\begin{cases}1,\ \textit{if}\ w_k\xi_k^{\mathsf{lo}}>\xi,\forall l_k\ st\ s_{k,l_k}=1\\ 0,\ \textit{otherwise}\end{cases}. \tag{1.10}$$

and $(\mathcal{P}_3)_k$'s are the subproblems that can be solved independently by individual UEs $(k\in\mathcal{K}_\xi)$ for given values of ξ and $|\mathcal{K}_\xi|$, which are given as

$$\begin{aligned}(\mathcal{P}_3)_k\quad &\min_{\boldsymbol{s}_k,f_k^{\mathsf{c}},p_k}\quad [f_k^{\mathsf{c}}]^+\\ &s.\ t.\quad (1.7b)_k,(1.7d)_k,(1.7e)_k,(1.7h)_k,(1.8a)_k,\end{aligned}$$

where $[f_k^{\mathsf{c}}]^+=\max(f_k^{\mathsf{c}},0)$, constraints $(1.7b)_k$, $(1.7d)_k$, $(1.7e)_k$, $(1.7h)_k$, and $(1.8a)_k$ denote the corresponding constraints $(1.7b)$, $(1.7d)$, $(1.7e)$, $(1.7h)$, and $(1.8a)$ for UE k, respectively.

Using the results in *Proposition 1.2*, we propose an optimal algorithm to solve problem $(\mathcal{P}_2)$ as described in Algorithm 1.1. Besides, for given $\boldsymbol{s}_k$, $(\mathcal{P}_3)_k$ is equivalently transformed to the

standard convex problem as shown in Section 5.4.1. Then, considering all different combinations of s_{k,l_k} ($l_k \in \mathcal{L}_k$), where $\boldsymbol{s}_k$ satisfies $(1.8a)_k$ and $(1.7e)_k$, we can determine $f_{k,\xi}^{\mathsf{c,min}}$ as well as the solution of $(\mathcal{P}_3)_k$.

b) P-CSI - Low-complexity Algorithm (P-SO)

We propose a low-complexity algorithm which iteratively solves two subproblems decomposed from problem $(\mathcal{P}_2)$ where the first one, i.e., the offloading optimization (OP) subproblem, determines offloading decision and computing resource allocation while the second one, i.e., the power allocation (PA) subproblem, performs uplink power allocation and reassigns the computing resource. First, for a given value of $\boldsymbol{p}$, the (OP) subproblem is given as follows:

$$(\mathcal{P}_2^{\mathsf{OP}}) \min_{\boldsymbol{S},\boldsymbol{f}^{\mathsf{c}},\xi} \xi \quad \text{s.t} \quad (1.7b), (1.7d) - (1.7f), (1.8a).$$

Second, with the offloading solution $\boldsymbol{S}$ obtained by solving $(\mathcal{P}_2^{\mathsf{OP}})$, the (PA) is given as

$$(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}) \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\xi} \xi \quad \text{s.t} \quad (1.7b), (1.7d), (1.7f), (1.7h).$$

The proposed algorithm (P-SO), which iteratively solves $(\mathcal{P}_2^{\mathsf{OP}})$ and $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ until convergence, is described in Algorithm 1.2. Besides, this approach is the key for solving problem $(\mathcal{P}_2)$ in the IP-CSI scenario when the finding of optimal solution would be impossible. Note that $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$ is the optimal of subproblem $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ at iteration q. We describe how to solve (OP) and (PA) in the following.

In order to tackle $(\mathcal{P}_2^{\mathsf{OP}})$, we will apply the decomposition technique as employed in Section 1.2.1.2 to further decompose this subproblem into individual users' small-scale subproblems:

$$(\mathcal{P}_2^{\mathsf{OP}})_k \quad \min_{\boldsymbol{s}_k, f_k^{\mathsf{c}}} \quad [f_k^{\mathsf{c}}]^+ \text{s.t.} \quad (1.7b)_k, (1.7d)_k, (1.7e)_k, (1.8a)_k.$$

To determine minimum of f_k^{c} for a given ξ, denoted as $f_{k,\xi}^{\mathsf{c,min}}$, we apply bisection search on ξ as in Algorithm 1.1, except for some difference in step 4 and step 5 to find $f_{k,\xi}^{\mathsf{c,min}}$. Let $\boldsymbol{S}_k^{\mathsf{bi}} \in \mathbb{R}^{2^{|\mathcal{L}_k|} \times |\mathcal{L}_k|}$ denote the binary matrix whose rows represent all possible combinations of task offloading decisions

Algorithm 1.2. Low-complexity Algorithm - P-CSI (P-SO)

1: **Initialize**: choose ϵ, initial $p_k^{(0)} = p_{\max}/2, \forall k$.
2: **while** $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})} - \xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{PA}})} < \epsilon$ **do**
3: Assign $q = q + 1$;
4: Solve $\mathcal{P}_2^{\mathsf{OP}}$ to get $\boldsymbol{S}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})}$.
5: Solve $\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}$ to get $\boldsymbol{p}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$.
6: **end while**

of UE k. For example, $\boldsymbol{S}_k^{\mathsf{bi}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}^{\mathsf{t}}$ in case of $|\mathcal{L}_k| = 2$. Then, it can be verified that the minimum value of f_k^{c} for a given value of ξ can be computed as

$$\begin{aligned}
f_{k,\xi}^{\mathsf{c,min}} &= \min((LC0_k \odot LC2_k \odot L20_k)\backslash\{0\}), \\
LC0_k &= \mathbb{1}_{\xi \times \mathbf{1}_{2^{|\mathcal{L}_k|}} - LC0'_k \geq 0}, \\
LC0'_k &= w_k \alpha_k \eta_k^{-2} (\boldsymbol{S}_k^{\mathsf{bi}} \boldsymbol{c}_k)^3 + \frac{w_k (p_k + p_{k,0})(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{b}_k}{r_k^{\mathsf{lb}}}, \\
LC2_k &= \left[(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{c}_k \oslash (\eta_k \times \mathbf{1}_{2^{|\mathcal{L}_k|}} - \frac{(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{b}_k}{r_k^{\mathsf{lb}}}) \right]^+, \\
L20_k &= \mathbb{1}_{\eta_k F_k - \boldsymbol{S}_k^{\mathsf{bi}} \boldsymbol{c}_k \geq 0},
\end{aligned} \tag{1.12}$$

where $\odot$ and $\oslash$ denote the Hadamard product and division, respectively, $\mathbf{1}_n$ represents the $n \times 1$ vector of ones, $\mathbb{1}_{\boldsymbol{x} \geq 0}$ is the indicator function and $\boldsymbol{x}^+ = \max(\boldsymbol{x}, 0)$. In above expressions, the elements of $LC0_k$ and $L20_k$ will be equal to 1 if the corresponding row of $\boldsymbol{S}_k^{\mathsf{bi}}$ satisfies constraint $(1.7b)_k$ and $(1.8a)_k$, respectively. The vector of $LC2_k$ describes the minimum value of f_k^{c} corresponding to each row of $\boldsymbol{S}_k^{\mathsf{bi}}$. It is noted that $LC0'_k, LC2_k$ and $L20_k$ do not depend on the value of ξ; thus, we just need to compute them at the beginning of the bisection search (the 'while-loop' in Algorithm 1.1) and use them to update $LC0_k$ and $f_{k,\xi}^{\mathsf{c,min}}$ corresponding to the updated value of ξ.

With the solution of $(\mathcal{P}_2^{\mathsf{OP}})$, we can then solve the $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ subproblem to obtain the optimal solutions of transmit power $\boldsymbol{p}_k$ and computing resource allocation $\boldsymbol{f}^{\mathsf{c}}$. This can be fulfilled by using a similar process employed in section 1.2.1.2 which applies the bisection search on ξ and solving subproblem $(\mathcal{P}_3)'_k$. It is noted that Algorithm 1.2 creates a sequence of feasible solutions for $(\mathcal{P}_2)$ where objective function value of this problem monotonically decreases over iterations.

1.2.1.3 Algorithm Design for IP-CSI Scenario

We tackle problem $(\mathcal{P}_2)$ by iteratively solving two subproblems $(\mathcal{P}_2^{\mathsf{OP}})$ and $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ until convergence. Because the IP-CSI only affects the transmission energy and transmission time, the (OP) subproblem $(\mathcal{P}_2^{\mathsf{OP}})$ can be solved as in Section 1.2.1.2, we only need to consider the (PA) subproblem $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$, which can be written as

$$\begin{aligned}(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}}) \quad & \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}}} \quad \xi \\ & \text{s. t.} \quad (1.7b): \ w_k(\xi_{k,\mathsf{IP}}^{\mathsf{t,ub}} + \xi_k^{\mathsf{lo}}) \le \xi, \\ & \qquad\quad (1.7d): \ t_{k,\mathsf{IP}}^{\mathsf{t,ub}} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} \le \eta_k, \ \ (1.7f), (1.7h).\end{aligned}$$

It can be verified that $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ is non-convex and NP-hard. To tackle it, we first apply data compression to approximately convexify the non-convex constraints (1.7b) and (1.7d) for a given value of ξ as follows

$$p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}} \tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \ \le \ 0, \tag{1.13}$$

$$\left(\frac{T}{T-\tau}\right) \frac{b_k^{\mathsf{a}}}{\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} - \eta_k \ \le \ 0. \tag{1.14}$$

where

$$\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) = W\log_2(p_k + \boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - W\log_2((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - \frac{\boldsymbol{\lambda}_k(\boldsymbol{p} - \boldsymbol{p}^{(q)})}{\log(2)\Big((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\Big)}, \tag{1.15}$$

and $\boldsymbol{p}^{(q)}$ is the point that $\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) = \hat{r}_k^{\mathsf{lb}}(\boldsymbol{p})$. It is noted that $\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \le \hat{r}_k^{\mathsf{lb}}(\boldsymbol{p})$ due to the fact that $v_k(\boldsymbol{p}) \le \tilde{v}_k(\boldsymbol{p}) = v_k(\boldsymbol{p}^{(q)}) + \nabla v_k(\boldsymbol{p}^{(q)})(\boldsymbol{p} - \boldsymbol{p}^{(q)})$ at point $\boldsymbol{p}^{(q)}$, where $v_k(\boldsymbol{p}) = W\log_2\Big(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\Big)$.

From (1.15), the feasibility verification of $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ for a given value of ξ is now equivalent to find at least one point $(\boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}, \boldsymbol{p}^{(q)})$ that makes constraints (1.7f) and (1.7h) and inequalities (1.13), (1.14) feasible. Toward this end, we will iteratively update $\boldsymbol{p}^{(q)}$ to make the approximation in (1.15)

Algorithm 1.3. PA Feasibility Verification - IP-CSI

1: **Initialize**: choose $\boldsymbol{p}^{(0)}$ as the previous solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$.
2: **repeat**
3: $q = q + 1$;
4: At $\boldsymbol{p} = \boldsymbol{p}^{(q-1)}$, solve $(\mathcal{P}^{\mathsf{PA}}_{2})^{(q-1)}$ to get $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$
5: **if** $\chi < 0$ **then**
6: Assign $feasibility = true$
7: Return $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$; **break**;
8: **else**
9: Assign $feasibility = false$
10: Compute $\tilde{r}^{\mathsf{lb}}_{k}(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ for all k
11: **end if**
12: **until** convergence

tighter and find the minimum χ for a given $\boldsymbol{p}^{(q)}$, where χ is the objective of the following problem:

$$
\begin{aligned}
(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)} \quad & \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\chi} \quad \chi && \text{(1.16a)}\\
& \text{s. t.} \quad p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}}\tilde{r}^{\mathsf{lb}}_{k}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq \chi, && \text{(1.16b)}\\
& \quad \left(\frac{T}{T-\tau}\right)\frac{b^{\mathsf{a}}_{k}}{\tilde{r}^{\mathsf{lb}}_{k}(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c^{\mathsf{a}}_{k}}{f^{\mathsf{c}}_{k}} - \eta_k \leq \chi, && \text{(1.16c)}\\
& \quad (1.7f), (1.7h),
\end{aligned}
$$

As shown above, the convexity of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)}$ is guaranteed; therefore, we can effectively solve this problem by using the CVX solver. Finally, the feasibility verification is presented in Algorithm 1.3, and the bisection search to solve $(\mathcal{P}^{\mathsf{PA}}_{2,\mathrm{IP}})$ is similar to Algorithm 1.1, except for the difference in step 5, where the feasibility verification is done as described in Algorithm 1.3. It can be verified that for a given ξ, using D.C to approximate the transmission rate (using the rate lower bound in (1.15)) and iteratively solving problem $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)}$ leads to convergence.

Proposition 1.3. *If the optimal value of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ is equal to zero at the convergence of Algorithm 1.3, then the solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ combining with ξ gives a stationary point of subproblem $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$, where Q denotes the final iteration index at the convergence of Algorithm 1.3.*

1.2.1.4 Consideration of Downlink Transmission

We extend the proposed design described in the previous sections to consider this downlink transmission. Let b^{dl}_{k,l_k} denote the number of downlink bits related to the computation result of task l_k,

which must be sent from the BS to UE k. Assuming that ZF precoder is employed, the lower-bound ergodic downlink rate can be expressed as

$$\hat{r}_k^{\mathsf{dl,lb}} = \begin{cases} W \log_2\left(1 + p_k^{\mathsf{dl}}/\sigma_k^{\mathsf{dl}}\right) & \text{(P-CSI)} \\ W \log_2\left(1 + \frac{p_k^{\mathsf{dl}}}{\sum\limits_{i \in \mathcal{K}_\xi} p_i^{\mathsf{dl}} \lambda_{k,i} + \sigma_k^{\mathsf{dl}}}\right) & \text{(IP-CSI)} \end{cases}, \tag{1.17}$$

where p_k^{dl} denote the power that BS uses to transmit the offloading result data to UE k and σ_k^{dl} represents the received noise power at UE k. Then, the constraint (1.7d) on the total latency can be expressed as

$$\frac{T}{T-\tau}\left(\frac{b_k^{\mathsf{a}}}{\hat{r}_k^{\mathsf{lb}}} + \frac{b_k^{\mathsf{a,dl}}}{\hat{r}_k^{\mathsf{dl,lb}}}\right) + \frac{c_k^{\mathsf{a}}}{f_k^c} \leq \eta_k, \quad \forall k \in \mathcal{K}_\xi, \tag{1.18}$$

where $b_k^{\mathsf{a,dl}} = \sum\limits_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}^{\mathsf{dl}}$. On the other hand, the average transmit power of BS can be computed as

$$p_{\mathsf{BS}} = \sum_{k \in \mathcal{K}_\xi} p_k^{\mathsf{dl}} \mathbb{E}(||\hat{\boldsymbol{a}}_{\boldsymbol{k}}||^2) = \begin{cases} \sum\limits_{k \in \mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}}{(M - |\mathcal{K}|_\xi)\beta_k}, & \text{P-CSI} \\ \sum\limits_{k \in \mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs})}{(M - |\mathcal{K}|_\xi)\tau p^{\mathsf{tr}} \beta_k^2}, & \text{IP-CSI} \end{cases}.$$

The total transmit power at BS must be constrained by its maximum power $p_{\mathsf{max}}^{\mathsf{dl}}$, which can be expressed as

$$p_{\mathsf{BS}} \leq p_{\mathsf{max}}^{\mathsf{dl}}. \tag{1.19}$$

We can now formulate the joint computation offloading and resource allocation problem considering both uplink and downlink data transmissions as follows:

$$(\mathcal{P}_2^{\mathsf{ext}}) \ \min \xi \quad \text{s.t} \quad (1.7b), (1.7e), (1.7f), (1.7h), (1.8a), (1.18), (1.19).$$

To tackle this difficult problem, we can again decompose it into two subproblems as in previous sections. In particular, we iteratively solve the (OP) subproblem (with constraints (1.7b), (1.7e), (1.7f), (1.8a), (1.18)) to find the optimal $\boldsymbol{s}, \boldsymbol{f}^{\mathsf{c}}$ and solve the extended (PA) subproblem (with constraints (1.7b), (1.7f), (1.7h), (1.18), (1.19)) to find $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$. Because the optimization variables $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$ are only captured in the extended (PA) subproblem, we can solve the (OP) subproblem as in section 1.2.1.2. To solve the extended (PA) subproblem, we can apply the same techniques as in section 1.2.1.3 to deal with the downlink rate, because the two delay components, which correspond to the uplink transmission time of the incurred data and download transmission time of the computation outcome, respectively, have the same structure.

1.2.1.5 Numerical Results

We consider an MEC system with the channel bandwidth of 10 MHz and $K = 20$ UEs randomly distributed in a cell coverage area with the radius of 900m. In our simulation setting, we set $F_k^{\mathsf{max}} = 2.4$ GHz, $P_k^{\mathsf{m}} = 0.22$ (Watts), $p_{k,0} = 0.05$, $\alpha_k = 0.1\times10^{-27}$, $|\mathcal{L}_k| = 5$, and $\eta_k = \eta$ for all k, $M = 30$, $F^{\mathsf{c}} = 40$ GHz, $p_{\mathsf{max}}^{\mathsf{dl}} = 10$ (Watts), $\sigma_{bs} = \sigma_k^{\mathsf{dl}} =$ bandwidth $\times 3.6 \times 10^{-21}$, $T = 200$ symbols. All UEs have the same number of parallel tasks and the same total computation demand of 0.24 Gcycles, but the number of CPU cycles per task is set randomly. The total number of transmission bits for all tasks is set to be the same for all UEs while the number of bits per task is generated randomly. For performance evaluation of the proposed design, we choose the ratio between the total number of transmission bits and the total required CPU cycles (BPC) to be about 4.2×10^{-3} (except for the results in Fig. 1.1), which is close to its highest possible value for the applications considered in [26]. The small scale channel fading coefficient is generated according to the Rayleigh distribution and the path-loss is defined according to 3GPP technical report as β_k (dB) $= 128.1 + 37.6\log_{10}(d_k)$ where d_k is the geographical distance between UE k and the BS (in km) [27].

The benefit of joint optimization of radio and computing resource allocation in the computation offloading design is illustrated in Fig. 1.1 for varying maximum allowable delay η. In this figure, considering no downlink data transmission and P-CSI scenario, we compare the achievable performance in four scenarios: task processing at mobile devices ('No-offload'), partial offloading with optimal offloading decision and cloud-resource allocation with fixed transmit power for all UEs $p_k = p_{\max}/2$ and $p_k = p_{\max}$, and with optimal transmit power allocation ('Optimal p_k'). The left

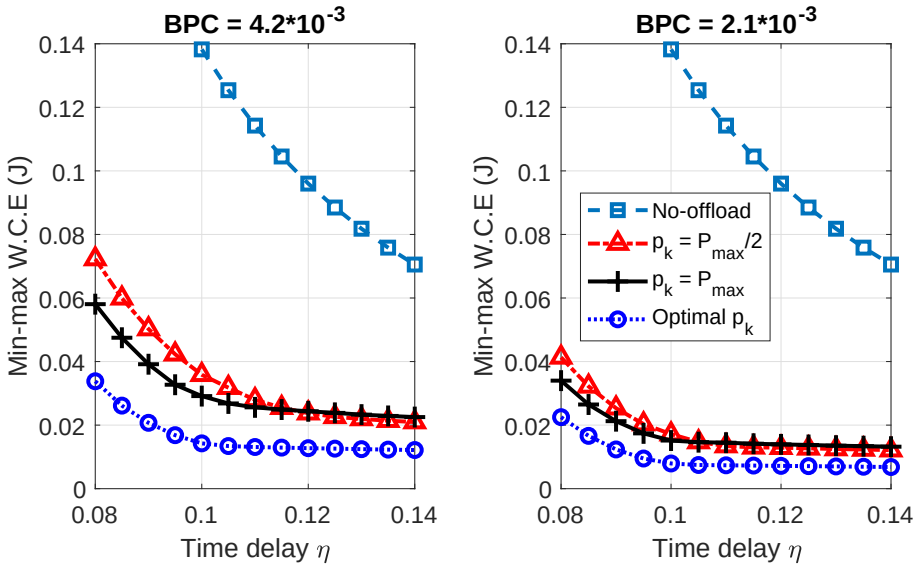

Figure 1.1 – Performance comparison of with/without offloading and with/without optimization of radio and computing resource.

and right subfigures show the achieved min-max W.C.E for different values of transmission bits per CPU cycle (BPC). From this figure, we can see that the minimum required latency that the mobile device can process its tasks locally ('No-offload' case) is 0.1s while the minimum required latency in the remaining cases are 0.08s. This means that computation offloading allows mobile devices to achieve lower latency. Moreover, the consumed energy in the partial offloading scheme is significantly smaller than that in the 'No-offload' case. For instance, the min-max W.C.E at $\eta = 0.1s$ in the left subfigure is equal to 0.138, 0.036, 0.029, 0.014 for the 'No-offload', fixed transmit power of '$p_k = p_{\max}/2$', '$p_k = p_{\max}$' and 'Optimal p_k', respectively. This means that partial offloading enables saving about 5 times of energy with no optimization of the transmit power and save about 10 times of energy with optimal transmit power. Moreover, the difference in the consumed energy among the offloading and no-offloading schemes becomes larger for smaller number of transmission bits.

Fig. 1.2 presents the achieved performance of different design scenarios considered in this paper: optimal solution with P-CSI - no downlink data ('P-O, noDL'), solution with P-CSI - no downlink data ('P-SO, noDL'), and IP-CSI - no downlink data ('IP-SO, noDL'). We also consider different application scenarios with small, medium and large amount of downlink data in comparison with amount of uplink data where the performance of our low-complexity algorithm for the IP-CSI scenario is investigated. Specifically, we set the ratio between the amount of downlink data and the amount of uplink data (Γ_{dpu}) (i.e., computed as $\sum_{l_k \in \mathcal{L}_k} b^{\mathsf{dl}}_{k,l_k} / \sum_{l_k \in \mathcal{L}_k} b_{k,l_k}$) equal to 0.5, 1, and

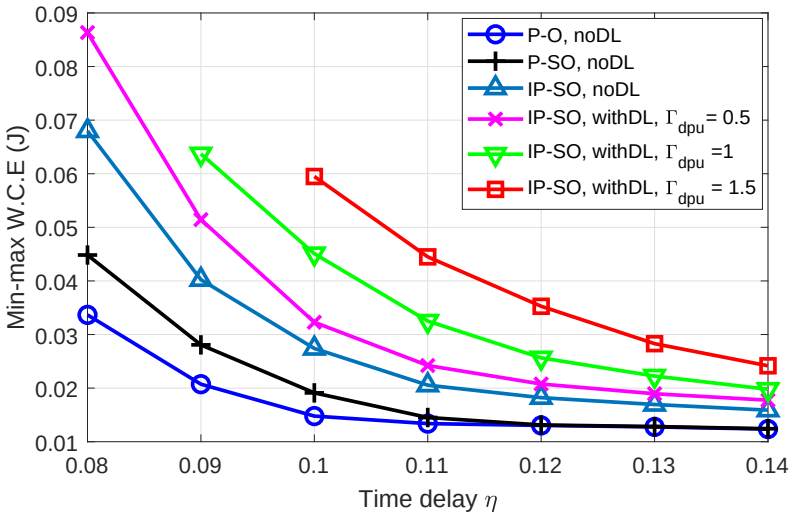

Figure 1.2 – Min-max W.C.E versus maximum allowable latency.

1.5 corresponding to low downlink data ('$\Gamma_{\mathsf{dpu}} = 0.5$'), medium downlink data ('$\Gamma_{\mathsf{dpu}} = 1$'), and large downlink data ('$\Gamma_{\mathsf{dpu}} = 1.5$'), respectively. It can be observed from this figure that the low-complexity algorithm achieves close-to-optimal performance when the maximum delay constraint is less stringent ('blue' and 'black' curves). For the IP-CSI scenario, mobile users will require more energy for data transmission to compensate for the CSI estimation errors. When the amount of downlink data becomes larger, more time is required to transfer the download data which means that less time is available for uploading the uplink data and computation at the cloud server. In some cases, increasing the transmit power to its maximum value may not lead to improved SINR, and the low transmission rate may prevent successful uplink data transmission in the offloading process. In all studied scenarios, even for the high value of Γ_{dpu}, the partial offloading scheme enables us to save energy significantly.

1.2.2 Joint Data Compression and Computation Offloading in Hierarchical Fog-Cloud Systems

To the best of our knowledge, the joint design of data compression, computation offloading, and resource allocation for hierarchical fog-cloud systems has not been considered in the existing literature. The main contributions of our work can be summarized as follows:

- We propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by data compression and decompression.

- For data compression at only the mobile users, we formulate the fair joint design of the compression ratio, computation offloading, and resource allocation as a MINLP optimization problem to minimize the maximum weighted energy and service delay cost (WEDC) of all users. We propose an optimal algorithm, referred to as Joint Data Compression, Computation Offloading, and Resource Allocation (JCORA) algorithm, which solves this challenging problem optimally.

- We then study a more general design where data compression is performed at both the mobile users and the fog server (with different compression ratios). We propose three different solution algorithms, namely adopted Piece-wise Linear Approximation (PLA) algorithm, One-dimensional λ-Search based Two-Stage (OSTS) algorithm, and Iterative λ-Update based Two-Stage (IUTS) algorithm, to solve this more general problem.

1.2.2.1 System Model

We consider a hierarchical fog-cloud system consisting of K single-antenna mobile users, one cloud server, and one fog server co-located with a base station (BS) equipped with a large number of antennas. In this system, the BS communicates with the users through wireless links while a (wired) backhaul link is deployed between the BS co-located with the fog server and the cloud server. For convenience, we denote the set of users as $\mathcal{K}$. We assume that each user k needs to execute an application requiring c_k CPU cycles within an interval of T_k^{max} seconds, where $c_{k,0}$ CPU cycles must be executed locally at the mobile device and the remaining offloadable $c_{k,1}$ CPU cycles can be processed locally or offloaded and processed at the fog/cloud server for energy saving and delay improvement. Let b_k^{in} be the number of bits representing the corresponding incurred data of the possibly-offloaded $c_{k,1}$ CPU cycles. Once $c_{k,1}$ CPU cycles are offloaded, user k first compresses the corresponding b_k^{in} bits down to $b_k^{\mathsf{out,u}}$ bits before sending them to the remote fog server. The compression ratio is denoted as $\omega_k^{\mathsf{u}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,u}}$.

a) Data Compression Model: We adopt a practical data-fitting approach to model the compression computational load, decompression computational load, and compression quality as

non-linear functions of the compression ratio as follows:

$$c_k^{\mathsf{x,u}} = \gamma_{k,0}^{\mathsf{u}}\left[\gamma_{k,1}^{\mathsf{x,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{x,u}}} + \gamma_{k,3}^{\mathsf{x,u}}\right],\ \text{for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{1.20}$$

$$q_k^{\mathsf{qu,u}} = \gamma_{k,3}^{\mathsf{qu,u}} - \left[\gamma_{k,1}^{\mathsf{qu,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{qu,u}}}\right],\ \text{for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{1.21}$$

where 'x' = 'co' and 'de' stands for compression and decompression, respectively, $[\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}]$ represents the possible range of ω_k^{u} and depends on the compression algorithm employed at user k, $c_k^{\mathsf{co,u}}$ and $c_k^{\mathsf{de,u}}$ denote the additional CPU cycles at source and destination needed for compression and decompression, respectively; $q_k^{\mathsf{qu,u}}$ represents the perceived QoS (i.e., this parameter, which is only considered for lossy compression, measures the deviation between the true data and the decompressed data); $\gamma_{k,0}^{\mathsf{u}}$ is the maximum number of CPU cycles; $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i = 1, 2, 3$, are constant parameters where $\gamma_{k,1}^{\mathsf{co/de/qu,u}}, \gamma_{k,3}^{\mathsf{co/de/qu,u}} \geq 0$.

b) Computing and Offloading Model: We now introduce the binary offloading decision variables s_k^{u}, s_k^{f}, and s_k^{c} for the computation task of user k, where $s_k^{\mathsf{u}} = 1$, $s_k^{\mathsf{f}} = 1$, and $s_k^{\mathsf{c}} = 1$ denote the scenarios where the application is executed at the mobile device, the fog server, and the cloud server, respectively; and these variables are zero otherwise. Moreover, we assume that the $c_{k,1}$ CPU cycles can be executed at exactly one location, which implies $s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 1$. Then, the total computational load of user k at the mobile device, denoted as c_k^{u}, and at the fog server, denoted as c_k^{f}, are given as, respectively, $c_k^{\mathsf{u}} = c_{k,0} + s_k^{\mathsf{u}} c_{k,1} + (1 - s_k^{\mathsf{u}}) c_k^{\mathsf{co,u}}$ and $c_k^{\mathsf{f}} = s_k^{\mathsf{f}}\left(c_{k,1} + c_k^{\mathsf{de,u}}\right)$.

The local computation energy consumed by user k and the local computation time can be expressed, respectively, as $\xi_{1,k}^{\mathsf{u}} = \alpha_k {f_k^{\mathsf{u}}}^2 c_k^{\mathsf{u}}$ and $t_{1,k}^{\mathsf{u}} = c_k^{\mathsf{u}}/f_k^{\mathsf{u}}$, where f_k^{u} is the CPU clock speed of user k and α_k denotes the energy coefficient specified by the CPU model [28]. Let f_k^{f} denote the CPU clock speed used at the fog server to process $c_{k,1}$. Then, the computing time at the fog server is given by $t_{1,k}^{\mathsf{f}} = c_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. We assume that the computation task of each user is executed at the cloud server with a fixed delay of T^{c} seconds.

c) Communication Model: We assume that channel estimation is perfect and ZF is applied at the BS, then the average uplink rate from user k to the BS (fog server) is expressed as $r_k = \rho_k \log_2(1 + P_k \beta_{k,0})$, where P_k is the uplink transmit power per Hz of user k, ρ_k denotes the transmission bandwidth, and $\beta_{k,0} = M_0 \beta_k / \sigma_{\mathsf{bs}}$. Here, β_k represents the large-scale fading coefficient, σ_{bs} is the noise power density (watts per Hz), and M_0 is the multiple-input multiple-output (MIMO)

beamforming gain [25]. It is assumed that the number of antennas is sufficiently large so that M_0 is identical for all users. Then, the uplink transmission time and energy of user k can be computed, respectively, as $t^{\mathsf{u}}_{2,k} = (1 - s^{\mathsf{u}}_k) b^{\mathsf{out,u}}_k / r_k$ and $\xi^{\mathsf{u}}_{2,k} = \rho_k (P_k + P_{k,0}) t^{\mathsf{u}}_{2,k}$, where $P_{k,0}$ denotes the circuit power consumption per Hz. For the data transmission between the fog server and the cloud server, a backhaul link with capacity D^{max} bps (bits per second) is assumed. Let d_k denote the backhaul rate allocated to user k. Then, the transmission time from the fog server to the cloud server is $t^{\mathsf{f}}_{2,k} = s^{\mathsf{c}}_k b^{\mathsf{out,u}}_k / d_k$.

Then, the total delay for completing the computation task of user k is given by

$$T_k = t^{\mathsf{u}}_{1,k} + t^{\mathsf{u}}_{2,k} + t^{\mathsf{f}}_{1,k} + t^{\mathsf{f}}_{2,k} + s^{\mathsf{c}}_k T^{\mathsf{c}}. \tag{1.22}$$

Furthermore, the overall energy consumed at user k for processing its task is given by

$$\xi_k = \xi^{\mathsf{u}}_{1,k} + \xi^{\mathsf{u}}_{2,k}. \tag{1.23}$$

Practically, all users want to save energy and enjoy low application execution latency. Hence, we adopt the WEDC as the objective function of each user k as $\Xi_k = w^{\mathsf{T}}_k T_k + w^{\mathsf{E}}_k \xi_k$, where w^{T}_k and w^{E}_k represent the weights corresponding to the service latency and consumed energy, respectively. These weights can be pre-determined by the users to reflect their priorities or interests. The proposed design aims to minimize the WEDC function for each user while maintaining fairness among all

Algorithm 1.4. Optimal Joint DC, Offloading, and Resource Allocation (JCORA)

1: **Initialize**: Compute $\eta_k^{\mathsf{lo}}, \forall k \in \mathcal{K}$ as in (1.25), choose ϵ, assign $\eta^{\mathsf{min}} = 0$, $\eta^{\mathsf{max}} = \max_k(\eta_k^{\mathsf{lo}})$, and set BOOL = *False*.
2: **while** $(\eta^{\mathsf{max}} - \eta^{\mathsf{min}} > \epsilon)$ & (BOOL = *False*) **do**
3: Assign $\eta = (\eta_{\mathsf{max}} + \eta_{\mathsf{min}})/2$, and then define sets $\mathcal{A} = \{k|\eta_k^{\mathsf{lo}} \leq \eta\}$ and $\mathcal{B} = \mathcal{K}/\mathcal{A}$.
4: Check feasibility of $(\mathcal{P}_{\mathcal{B}})$ as in Section 1.2.2.2.
5: **if** $(\mathcal{P}_{\mathcal{B}})$ *is feasible* **then** $\eta^{\mathsf{max}} = \eta$, BOOL = *True*, **else** $\eta^{\mathsf{min}} = \eta$, BOOL = *False*, **end if**
6: **end while**

users. Towards this end, we consider the following min-max optimization problem:

$$
\begin{aligned}
(\mathcal{P}_2) \quad \min_{\Omega_1 \cup \eta} \quad & \eta && \text{(1.24a)}\\
\text{s. t.} \quad & \Xi_k \leq \eta, \forall k, && \text{(1.24b)}\\
& f_k^{\mathsf{u}} \leq F_k^{\mathsf{max}}, \forall k, && \text{(1.24c)}\\
& \sum_k f_k^{\mathsf{f}} \leq F^{\mathsf{f,max}}, && \text{(1.24d)}\\
& s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}} \in \{0,1\}, \forall k, && \text{(1.24e)}\\
& s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 1, \forall k, && \text{(1.24f)}\\
& \omega_k^{\mathsf{u,min}} \leq \omega_k^{\mathsf{u}} \leq \omega_k^{\mathsf{u,max}}, \forall k, && \text{(1.24g)}\\
& 0 \leq \rho_k P_k \leq p_k^{\mathsf{max}}, \forall k, && \text{(1.24h)}\\
& 0 \leq \rho_k \leq \rho_k^{\mathsf{max}}, \ \forall k, && \text{(1.24i)}\\
& \textstyle\sum_k d_k \leq D^{\mathsf{max}}, , && \text{(1.24j)}\\
& T_k \leq T_k^{\mathsf{max}}, \forall k, && \text{(1.24k)}
\end{aligned}
$$

where $\Omega_1 = \cup_{k\in\mathcal{K}}\Omega_{1,k}$, $\Omega_{1,k} = \{s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, \omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, P_k, \ \rho_k, d_k\}$; F_k^{max} is the maximum CPU clock speed of user k, $F^{\mathsf{f,max}}$ is the maximum CPU clock speed of the fog server, p_k^{max} is the maximum transmit power of user k, $[\omega_k^{\mathsf{u,min}}, \omega_k^{\mathsf{u,max}}]$ denotes the feasible range of the compression ratio ω_k^{u} which can guarantee the required QoS of the recovered data, ρ_k^{max} is the maximum constrained service cost. In particular, for lossless data compression where the perceived QoS $q_k^{\mathsf{qu,u}} = 1$ for all ω_k^{u}, this feasible range is determined as $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ and $\omega_k^{\mathsf{u,max}} = \omega_{k,1}^{\mathsf{u,max}}$. For lossy data compression where the perceived QoS is required to be greater than $q_k^{\mathsf{qu,u,min}}$, this range is determined as $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ and $\omega_k^{\mathsf{u,max}} = \min\left\{\omega_{k,1}^{\mathsf{u,max}}, \left((\gamma_{k,3}^{\mathsf{qu,u}} - q_k^{\mathsf{qu,u,min}})/\gamma_{k,1}^{\mathsf{qu,u}}\right)^{1/\gamma_{k,2}^{\mathsf{qu,u}}}\right\}$.

Let η_k^{lo} be the minimum WEDC due to UE k when it executes its computation task locally, which is computed from $(\mathcal{P}_2)$ as

$$\eta_k^{\mathsf{lo}} = \begin{cases} \mathcal{Q}_{k,0}(f_k^{\mathsf{u,sta}}), \text{ if } f_k^{\mathsf{u,sta}} \in [f_k^{\mathsf{u,min}}, F_k^{\mathsf{max}}] \\ \min\Big(\mathcal{Q}_{k,0}(f_k^{\mathsf{u,min}}), \mathcal{Q}_{k,0}(F_k^{\mathsf{max}})\Big), \text{ otherwise,} \end{cases} \tag{1.25}$$

where $\mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) = w_k^{\mathrm{E}}\alpha_k(f_k^{\mathsf{u}})^2 c_k + w_k^{\mathrm{T}} c_k / f_k^{\mathsf{u}}$, and $f_k^{\mathsf{u,min}} = c_k / T_k^{\mathsf{max}}$ and $f_k^{\mathsf{u,sta}} = \sqrt[3]{w_k^{\mathsf{T}}/(2w_k^{\mathsf{E}}\alpha_k)}$.

It can be verified that if $\eta^\star$ is the optimum objective value of problem $(\mathcal{P}_2)$, then an optimal classification, $(\mathcal{A}^\star, \mathcal{B}^\star)$, can be determined as $\mathcal{A}^\star = \{k | \eta_k^{\mathsf{lo}} \leq \eta^\star\}$, and $\mathcal{B}^\star = \mathcal{K}\backslash\mathcal{A}^\star$, where $\mathcal{A}$ and $\mathcal{B}$ be the locally executing and the offloading user sets, respectively. Therefore, we propose Algorithm 1.4, named as JCORA, to tackle $(\mathcal{P}_2)$. In this algorithm, we initially calculate η_k^{lo} for all users in $\mathcal{K}$ as in (1.25). Then, we employ the bisection search to find the optimum $\eta^\star$ where upper bound η^{max} and lower bound η^{min} are iteratively updated until the difference between them becomes sufficiently small, $(\mathcal{P}_\mathcal{B})$ is feasible, and the sets $\mathcal{A}$ and $\mathcal{B}$ do not change. It is noted that $(\mathcal{P}_\mathcal{B})$ is $(\mathcal{P}_2)$ for the users in set $\mathcal{B}$. The feasibility verification of $(\mathcal{P}_\mathcal{B})$ is presented as follows.

1.2.2.2 Feasibility Verification of $(\mathcal{P}_\mathcal{B})$

In order to verify the feasibility of $(\mathcal{P}_\mathcal{B})$, we consider the following problem

$$(\mathcal{P}_{\mathsf{FV},\eta}) \quad \min_{\Omega_\mathcal{B}} \sum_{k\in\mathcal{B}} f_k^{\mathsf{f}} \quad \text{s. t.} \quad (1.24b), (1.24c), (1.24e) - (1.24k).$$

This problem minimizes the total required computing resource of the fog server subject to all constraints of $(\mathcal{P}_\mathcal{B})$ except $(1.24d)$. Let $G^\star_{\mathcal{B},\eta}$ be the objective value of problem $(\mathcal{P}_{\mathsf{FV},\eta})$. Then, the feasibility of $(\mathcal{P}_\mathcal{B})$ can be verified by comparing $G^\star_{\mathcal{B},\eta}$ to the available fog computing resource $F^{\mathsf{f,max}}$. In particular, problem $(\mathcal{P}_\mathcal{B})$ is feasible if $G^\star_{\mathcal{B},\eta} \leq F^{\mathsf{f,max}}$. Otherwise, $(\mathcal{P}_\mathcal{B})$ is infeasible.

To solve $(\mathcal{P}_{\mathsf{FV},\eta})$, we consider two following subproblems

$$(\mathcal{P}_3)_k \quad \min_{\Omega_{2,k}} f_k^{\mathsf{f}} \quad \text{s. t.} \quad s_k^{\mathsf{f}} = 1,\ (1.24b)_k,\ (1.24c)_k,\ (1.24g)_k - (1.24i)_k,\ (1.24k)_k,$$

$$(\mathcal{P}_4)_k \quad \min_{\Omega_{2,k}\cup d_k\backslash f_k^{\mathsf{f}}} d_k \quad \text{s. t.} \quad s_k^{\mathsf{c}} = 1, (1.24b)_k,\ (1.24c)_k,\ (1.24g)_k - (1.24i)_k,\ (1.24k)_k,$$

Algorithm 1.5. Feasibility Verification of $(\mathcal{P}_\mathcal{B})$

Solve $(\mathcal{P}_3)_k$ to find $f_k^{\mathsf{f},\mathsf{rq}}, \forall k \in \mathcal{B}$.
Solve $(\mathcal{P}_4)_k$ to find $d_k^{\mathsf{rq}}, \forall k \in \mathcal{B}$.
if $\exists k$ such that $s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 0$ **then**
  Return $(\mathcal{P}_\mathcal{B})$ is infeasible
else
  Solve $(\mathcal{P}_{\mathsf{FV},\eta})$ to find $G^\star_{\mathcal{B},\eta}$.
  if $G^\star_{\mathcal{B},\eta} < F^{\mathsf{f},\mathsf{max}}$ **then** Return $(\mathcal{P}_\mathcal{B})$ is feasible, **else** Return $(\mathcal{P}_\mathcal{B})$ is infeasible **end if**
end if

where $\Omega_{2,k} = \{\omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, p_k, \rho_k\}$, $(1.24b)_k$, $(1.24c)_k$, $(1.24g)_k - (1.24i)_k$, and $(1.24k)_k$ denote the respective constraints of user k corresponding to $(1.24b)$, $(1.24c)$, $(1.24g) - (1.24i)$, and $(1.24k)$.

In sub-problems $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$, let $\tilde{\omega}_k^{\mathsf{u}} = \log(\omega_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{u}} = \log(f_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{f}} = \log(f_k^{\mathsf{f}})$, $\tilde{P}_k = \log(P_k)$, and $\tilde{\rho}_k = \log(\rho_k)$, we prove that $(\mathcal{P}_3)_k$ is convex with respect to set $\tilde{\Omega}_{2,k} \cup \tilde{l}_k$, where $\tilde{l}_k = \tilde{\omega}_k^{\mathsf{u}} + \tilde{\rho}_k$ and $\tilde{\Omega}_{2,k} = \{\tilde{\omega}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{f}}, \tilde{P}_k, \tilde{\rho}_k\}$. Similarly, $(\mathcal{P}_4)_k$ can be converted to a convex problem via logarithmic transformation. If $(\mathcal{P}_3)_k$ is infeasible, we set $s_k^{\mathsf{c}} = 0$, and if $(\mathcal{P}_4)_k$ is infeasible, we set $s_k^{\mathsf{c}} = 0$. With the obtained optimal objective of $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$, denoted $f_k^{\mathsf{f},\mathsf{rq}}$ and d_k^{rq}, respectively, the mode-mapping problem which is the equivalent of problem $(\mathcal{P}_{\mathsf{FV},\eta})$, is defined as

$$
\begin{aligned}
(\mathcal{P}_{\mathsf{FV},\eta}) \quad & \min_{\Omega_3} \ \mathcal{G}_{\mathcal{B},\eta}(\Omega_3) = \sum_{k\in\mathcal{B}} (1 - s_k^{\mathsf{c}}) f_k^{\mathsf{f},\mathsf{rq}} \\
& \text{s. t.} \ \sum_{k\in\mathcal{B}} s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \leq D^{\mathsf{max}}, \ s_k^{\mathsf{c}} \in \{0,1\},
\end{aligned}
$$

where $\Omega_3 = \{s_k^{\mathsf{c}} | k \in \mathcal{B}\}$ for a given η. In fact, $(\mathcal{P}_{\mathsf{FV},\eta})$ is a *"0-1 knapsack"* problem [29], which can be solved optimally and effectively using the CVX solver. If $G^\star_{\mathcal{B},\eta} \leq F^{\mathsf{f},\mathsf{max}}$, combining the set of all solutions of the $(\mathcal{P}_3)_k$'s, $(\mathcal{P}_4)_k$'s, and $(\mathcal{P}_{\mathsf{FV},\eta})$ yields a feasible solution of $(\mathcal{P}_\mathcal{B})$ for this value of η. Hence, $(\mathcal{P}_\mathcal{B})$ is feasible in such scenario. The feasibility verification of $(\mathcal{P}_\mathcal{B})$ is summarized in Algorithm 1.5.

Theorem 1.1. *The integration of Algorithm 1.5 into Algorithm 1.4 yields the global optimum of MINLP* $(\mathcal{P}_2)$.

Proof. Algorithm 1.5 verifies the feasibility of $(\mathcal{P}_\mathcal{B})$ for any given value of $\eta_\mathcal{B} = \eta$. Therefore, if Algorithm 1.4 employs Algorithm 1.5, $(\mathcal{P}_2)$ is solved optimally. Note that after convergence, the optimal variables are given by the optimal solution of $(\mathcal{P}_3)_k$ if $s_k^{\mathsf{f}} = 1$ or $(\mathcal{P}_4)_k$ if $s_k^{\mathsf{c}} = 1$ where the values of the s_k^{f}'s and s_k^{c}'s are the outcomes of $(\mathcal{P}_{\mathsf{FV},\eta})$. □

1.2.2.3 Data compression at Both Mobile Users and Fog Server

We now consider the more general case where the fog server also performs data compression before transmitting the compressed data over the backhaul link to the cloud server[2]. This design option can further enhance the performance for systems with a congested backhaul link. The backhaul compression ratio is defined as $\omega_k^{\mathsf{f}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,f}}$ where $b_k^{\mathsf{out,f}}$ stands for the number of bits transmitted over the backhaul link. Denote s_k^{m} as the binary variable indicating whether or not data compression is performed at the fog server for user k ($s_k^{\mathsf{m}} = 1$ for DC, and $s_k^{\mathsf{m}} = 0$, otherwise). In this general case, constraints (1.24e) and (1.24f) can be rewritten as

$$s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}} \in \{0,1\}, \forall k \in \mathcal{K}, \tag{1.26a}$$

$$s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + s_k^{\mathsf{m}} = 1, \forall k \in \mathcal{K}, \tag{1.26b}$$

Then, the computational load for compression and the output data corresponding to *Mode 3* can be modeled as $c_k^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}}\left[\gamma_{k,1}^{\mathsf{co,f}}(\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \gamma_{k,3}^{\mathsf{co,f}}\right]$ and $b_k^{\mathsf{out,f}} = b_k^{\mathsf{in}}/\omega_k^{\mathsf{f}}$, respectively, where $\gamma_{k,0}^{\mathsf{f}}, \gamma_{k,1}^{\mathsf{co,f}}, \gamma_{k,3}^{\mathsf{co,f}} \in \mathbb{R}_+$ are positive numbers. Here, we have additional constraints for the compression ratio at the fog server as

$$\omega_k^{\mathsf{f}} \in [\omega_k^{\mathsf{f,min}}, \omega_k^{\mathsf{f,max}}],\ \forall k \in \mathcal{K}. \tag{1.27}$$

Then, the total computational load for user k at the fog server becomes $\check{c}_k^{\mathsf{f}} = s_k^{\mathsf{f}}\left(c_{k,1} + c_k^{\mathsf{de,u}}\right) + s_k^{\mathsf{m}}(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})$, and the computing time at the fog server is $\check{t}_{1,k}^{\mathsf{f}} = \check{c}_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. Moreover, the transmission time incurred by offloading the data of user k from the fog server to the cloud server can be rewritten as $\check{t}_{2,k}^{\mathsf{f}} = \left(s_k^{\mathsf{f}} b_k^{\mathsf{out,u}} + s_k^{\mathsf{m}} b_k^{\mathsf{out,f}}\right)/d_k$. Then, the total delay for completing the computation task of user k is given by $\check{T}_k = t_{1,k}^{\mathsf{u}} + t_{2,k}^{\mathsf{u}} + \check{t}_{1,k}^{\mathsf{f}} + \check{t}_{2,k}^{\mathsf{f}} + (s_k^{\mathsf{c}} + s_k^{\mathsf{m}})T^{\mathsf{c}}$, and the WEDC becomes $\check{\Xi}_k = w_k^{\mathsf{T}}\check{T}_k + w_k^{\mathsf{E}}\xi_k$. Then, constraint (1.24k) is rewritten as

$$\check{T}_k \leq T_k^{\mathsf{max}}. \tag{1.28}$$

[2]It is assumed that the compression algorithms deployed in the mobile users are less efficient than the compression algorithms deployed in the fog server.

Algorithm 1.6. PLA-based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **Initialize**: L, η
2: Compute $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all $k \in \mathcal{B}$ as in Step 1 and 2 of Algorithm 1.5.
3: Define $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L,\ \forall k \in \mathcal{B}, l = 0 : L$.
4: Compute $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$. **If** $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ is unbounded **then** Remove point $d_{k,l}$ **end if**.
5: Compute $A_{k,l}$, $B_{k,l}$, and then solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$ to get optimal value $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star}$ of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$.
6: **if** $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star} \leq F^{\mathsf{f,max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible **end if**

The extended versions of problem $(\mathcal{P}_2)$ can be stated as

$$
\begin{aligned}
(\mathcal{P}_2^{\mathsf{ext}}) \quad & \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\} \cup \eta} \quad \eta \\
& \text{s. t.} \quad \breve{\Xi}_k \leq \eta, && (1.29a) \\
& \qquad (1.24c), (1.24d), (1.24g){-}(1.24j), (1.26a), (1.26b), (1.27), (1.28).
\end{aligned}
$$

To solve the extended problem, we employ the general solution approach presented in the previous Section. Now, we present the methods for the feasibility verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$.

a) Piece-wise Linear Approximation based Algorithm (PLA): After determining $f_k^{\mathsf{f,rq}}$ and d_k^{rq}, respectively, we determine the required fog computing resources for a given $d_k \in (0, d_k^{\mathsf{rq}})$ by solving the following problem:

$$
\begin{aligned}
(\mathcal{P}_{d_k}) \quad & \min_{\Omega_{2,k} \cup \{\omega_k^{\mathsf{f}}\}} f_k^{\mathsf{f}} \\
& \text{s. t. } s_k^{\mathsf{m}} = 1,\ (1.29a)_k,\ (1.24c)_k,\ (1.24g)_k - (1.24i)_k,\ (1.28)_k,\ (1.27)_k.
\end{aligned}
$$

Let $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ be the optimal solution of this problem, which can be obtained by employing the logarithmic transformations described in Section 1.2.2.2. However, finding a closed-form expression for $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ is not tractable. Hence, we propose to employ the *"Piece-wise Linear Approximation"* (PLA) method to divide the original domain into multiple small segments such that $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ can be approximated by a linear function in each segment. Suppose that the interval $[\epsilon_{\mathsf{d}}, d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}}]$ is divided into L segments of equal size, where ϵ_{d} is a very small number compared to d_k^{rq}, e.g, $\epsilon_{\mathsf{d}} = 1$. Specifically, the l^{th} segment corresponds to interval $[d_{k,l}, d_{k,l+1}]$, where $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L$ is a point such that $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ and the value of the approximated function at this point are equal. Then, we can approximate $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ as $\hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}\Big(V_k, U_k\Big) = \sum_{l=0}^{L-1} \left(v_{k,l} A_{k,l} + u_{k,l} B_{k,l}\right)$, where $A_{k,l} =$

$\Big(\mathcal{F}^{\mathsf{f},\mathsf{rq}}_{k,\eta}(d_{k,l+1}) - \mathcal{F}^{\mathsf{f},\mathsf{rq}}_{k,\eta}(d_{k,l})\Big)/(d_{k,l+1} - d_{k,l})$, $B_{k,l} = \mathcal{F}^{\mathsf{f},\mathsf{rq}}_{k,\eta}(d_{k,l}) - A_{k,l}d_{k,l}$, $V_k = \{v_{k,l}, l = 0, 1, ..., L-1\}$, $U_k = \{u_{k,l}, l = 0, 1, ..., L-1\}$, and continuous variable $v_{k,l}$ and binary variable $u_{k,l}$ satisfy the following constraints:

$$s^{\mathsf{m}}_k = \sum_{l=0}^{L-1} u_{k,l} \leq 1, \forall k \in \mathcal{B}, \tag{1.30}$$

$$u_{k,l}d_{k,l} \leq v_{k,l} \leq u_{k,l+1}d_{k,l+1}, \forall k \in \mathcal{B}, l = 0, 1, ..., L-1. \tag{1.31}$$

Then, we have $s^{\mathsf{m}}_k d_k = \sum_{l=0}^{L-1} v_{k,l}$. Therefore, problem $(\mathcal{P}_{\mathsf{FV},\eta})$, which is used to determine the minimum total required fog computing resources for all users, is modified in this extended case as follows:

$$(\mathcal{P}^{\mathsf{PLA}}_{\mathsf{FV},\eta}) \quad \min_{\check{\Omega}_3} \quad \hat{\mathcal{G}}^{\mathsf{PLA}}_{\mathcal{B},\eta}\left(\check{\Omega}_3\right) = \sum_{k\in\mathcal{B}} \left(s^{\mathsf{f}}_k f^{\mathsf{f},\mathsf{rq}}_k + \hat{\mathcal{F}}^{\mathsf{f},\mathsf{rq}}_{k,\eta}\left(V_k, U_k\right)\right)$$

$$\text{s. t.} \quad s^{\mathsf{f}}_k, s^{\mathsf{c}}_k, u_{k,l} \in \{0,1\}, \ \forall k, l, \tag{1.32a}$$

$$s^{\mathsf{f}}_k + s^{\mathsf{c}}_k + \sum\nolimits_{l=0}^{L-1} u_{k,l} = 1, \tag{1.32b}$$

$$u_{k,l}d_{k,l} \leq v_{k,l} \leq u_{k,l+1}d_{k,l+1}, \ \forall k, l, \tag{1.32c}$$

$$\sum\nolimits_{k\in\mathcal{B}} \left(\sum\nolimits_{l=0}^{L-1} v_{k,l} + s^{\mathsf{c}}_k d^{\mathsf{rq}}_k\right) \leq D^{\mathsf{max}}, \tag{1.32d}$$

where $\check{\Omega}_3 = \cup_{k\in\mathcal{B}}\left(s^{\mathsf{f}}_k \cup s^{\mathsf{c}}_k \cup U_k \cup V_k\right)$ and constraints (1.32a), (1.32b), and (1.32c)-(1.32d) are the transformed constraints of original constraints (1.26a), (1.26b), and (1.24j), respectively. This transformed problem is an MILP problem, which can be solved effectively by using the CVX solver. The PLA based algorithm for verifying the feasibility of $(\mathcal{P}^{\mathsf{ext}}_{\mathcal{B}})$ is summarized in Algorithm 1.6, which can be integrated into Algorithm 1.4 to solve $(\mathcal{P}^{\mathsf{ext}}_2)$. It is noted that if the value of $\mathcal{F}^{\mathsf{f},\mathsf{rq}}_{k,\eta}(d_{k,l})$ is unbounded for a given $d_{k,l}$, this infeasible point is removed when applying the PLA based algorithm.

b) Two-stage Solution Approach (TSA)

In this section, two two-stage algorithms are developed by exploiting the fact that the decompression computational load (and therefore, the associated energy consumption) is almost independent from the compression ratio. This implies that for a given η, the optimal values f^{u}_k, ω^{u}_k, p_k, and ρ_k for mobile user k are similar for both $s^{\mathsf{f}}_k = 1$ and $s^{\mathsf{c}}_k = 1$. Hence, in the first stage, after solving $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$, $\forall k \in \mathcal{B}$, we can set these variables to the corresponding optimal solution of $(\mathcal{P}_3)_k$, denoted as $f^{\mathsf{u}\star}_{k,1}$, $\omega^{\mathsf{u}\star}_{k,1}$, $p^{\star}_{k,1}$, and $\rho^{\star}_{k,1}$. In the second stage, we find the remaining variables pertaining

to the fog server $\Omega_4 = \cup_{k\in\mathcal{B}}\{s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}}, d_k, f_k^{\mathsf{f}}, \omega_k^{\mathsf{f}}\}$ by solving the following problem[3]:

$$
\begin{aligned}
(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}}) \quad \min_{\Omega_4} \quad & \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{TSA}}(\Omega_4) = \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{m}} f_k^{\mathsf{f}} + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \right) \\
\text{s. t.} \quad & s_k^{\mathsf{m}} \left(\frac{b_k^{\mathsf{out,f}}}{d_k} + \frac{(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})}{f_k^{\mathsf{f}}} \right) \leq \nu_{k,0}, \qquad (1.33a) \\
& \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{m}} d_k + s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \right) \leq D^{\mathsf{max}}, \qquad (1.33b) \\
& (1.26a), (1.26b), (1.27),
\end{aligned}
$$

where $\nu_{k,0} = \min\{(\eta - \Xi_{k,1})/w_k^{\mathsf{T}}, T_k^{\mathsf{max}} - T_{k,1}\} + (c_{k,1} + c_k^{\mathsf{de}})/f_k^{\mathsf{f,rq}} - T^{\mathsf{c}}$, and $\Xi_{k,1}$ and $T_{k,1}$ are the optimal values of Ξ_k and T_k in $(\mathcal{P}_3)_k$, respectively. Because $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ is a difficult MINLP problem, we tackle it by reducing the set of variables based on the following observations.

Observation 1: For any value of d_k's satisfying (1.33b), the optimal solution of f_k^{f} in $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ can be determined as $f_k^{\mathsf{f}\star} = s_k^{\mathsf{m}} \frac{(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})}{\nu_{k,0} - b_k^{\mathsf{out,f}}/d_k} = s_k^{\mathsf{m}} \mathcal{H}_0\Big(\omega_k^{\mathsf{f}}, d_k\Big)$, where $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}}, d_k\Big) = \frac{\omega_k^{\mathsf{f}} d_k \left[\tilde{\gamma}_{k,1}^{\mathsf{co,f}} (\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}} \right]}{\nu_{k,0}\omega_k^{\mathsf{f}} d_k - b_k^{\mathsf{in}}}$, $\tilde{\gamma}_{k,1}^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}} \gamma_{k,1}^{\mathsf{co,f}}$, and $\tilde{\gamma}_{k,3}^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}} \gamma_{k,3}^{\mathsf{co,f}} + c_k^{\mathsf{de,u}}$.

Observation 2: When $s_k^{\mathsf{m}} = 1$ and $d_k \geq \bar{d}_{k,1}$, the optimal value of ω_k^{f}, denoted as $\omega_k^{\mathsf{f}\star}$, is given as follows:

$$
\omega_k^{\mathsf{f}\star} = \begin{cases} \omega_k^{\mathsf{max,f}}, \text{ if } \gamma_{k,2}^{\mathsf{co,f}} \leq 0 \cup \{\gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,1} < d_k \leq \bar{d}_{k,2}\}, \\ \mathrm{inv}\Big(\mathcal{H}_1\Big(d_k\Big)\Big), \quad \text{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,2} < d_k \leq \bar{d}_{k,3}, \\ \omega_k^{\mathsf{f,min}}, \quad \text{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, d_k > \bar{d}_{k,3}, \end{cases} \qquad (1.34)
$$

where $\bar{d}_{k,1} = b_k^{\mathsf{in}}/(\nu_{k,0}\omega_k^{\mathsf{f}})$, $\bar{d}_{k,2} = \mathcal{H}_1\Big(\omega_k^{\mathsf{max,f}}\Big)$, $\bar{d}_{k,3} = \mathcal{H}_1\Big(\omega_k^{\mathsf{f,min}}\Big)$, and $\mathrm{inv}\Big(\mathcal{H}_1\Big(d_k\Big)\Big)$ is the value of ω_k^{f} for which $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)$ is equal to d_k, and $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big) \triangleq \frac{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} b_k^{\mathsf{in}} (\gamma_{k,2}^{\mathsf{co,f}} + 1) \Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}} b_k^{\mathsf{in}}}{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} \nu_{k,0} \gamma_{k,2}^{\mathsf{co,f}} \Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}} + 1}}$.

[3]We note that by reducing the number of optimization variables in $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$, the complexity of the resulting algorithms for feasibility verification of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is lower than that of the PLA based algorithm.

Algorithm 1.7. One-dimensional Search Based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **initialize**: Δ_λ, $\lambda = 0$, Assign $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible.
2: Define $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all k as in Step 2 and Step 3 of Algorithm 1.5.
3: **repeat**
4: Assign $\lambda = \lambda + \Delta_\lambda$. Compute $d_{k,\lambda}$ as in (1.35) and solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ to find $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$.
5: **if** $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \leq F^{\mathsf{f,max}}$ **then**
6: Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible; **break**
7: **end if**
8: **until** $\lambda = \lambda^{\mathsf{max}}$

Then, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ is equivalent to the following problem:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}}) \quad \min_{\tilde{\Omega}_4} \quad \sum_{k\in\mathcal{B}} \left[s_k^{\mathsf{m}} \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big) + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \right]$$
$$\text{s. t. } (1.26a), (1.26b), (1.33b),$$

where $\tilde{\Omega}_4 = \cup_{k\in\mathcal{B}}\{s_k^{\mathsf{c}}, s_k^{\mathsf{f}}, s_k^{\mathsf{m}}, d_k\}$. It can be verified that the optimal value of d_k for $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, denoted as $d_k^\star$, is given as follows:

$$d_k^\star = \begin{cases} 0, \quad \text{if} \quad s_k^{\mathsf{f}\star} = 1, \\ d_k^{\mathsf{rq}}, \quad \text{if} \quad s_k^{\mathsf{c}\star} = 1, \\ \left\{ d_{k,\lambda} \middle| \left(\frac{\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)}{\partial d_k}\Big|_{d_k = d_{k,\lambda}} \right) + \lambda = 0 \right\}, \text{otherwise}, \end{cases} \tag{1.35}$$

where λ is the Lagrange multiplier of constraint (1.33b).

Observation 3: The gradient $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k$ is a monotonically increasing function of d_k.

With **Observation 3**, we can conclude that for a given λ, there exists at most one value of d_k satisfying $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k + \lambda = 0$. This means if the optimal λ is known, problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ can be solved effectively. Therefore, as described in the following, to solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, we propose two algorithms: one is based on a one-dimensional search for λ, and the other one is based on iterative updating λ.

b1) One-dimensional λ-search based two-stage algorithm (OSTS Alg.) For a given λ, suppose that $d_{k,\lambda}$ satisfies $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k\Big|_{d_k = d_{k,\lambda}} + \lambda = 0$. By defining $f_{k,\lambda} = \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k = d_{k,\lambda}}$, $\mu_{k,\lambda} = s_k^{\mathsf{m}}$, $\mu_{k,\lambda} = 1 - s_k^{\mathsf{c}}$, and $\mu_{k,\lambda} = s_k^{\mathsf{c}}(1 - x_k)$, we can find the optimal solution of $\cup_{k\in\mathcal{B}}\{s_k^{\mathsf{c}}, x_k, d_k\}$

by solving the following problem:

$$(\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda \quad \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda) = \min_{\cup_{k\in\mathcal{B}} s_{k,\lambda}} \sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_{k,\lambda} f_{k,\lambda} + s^{\mathsf{f}}_{k,\lambda} f^{\mathsf{f,rq}}_k \right]$$
$$\text{s. t.} \quad \sum_{k\in\mathcal{B}} s^{\mathsf{m}}_{k,\lambda} d_{k,\lambda} + (1 - s^{\mathsf{f}}_{k,\lambda} - s^{\mathsf{m}}_{k,\lambda}) d^{\mathsf{rq}}_k \leq D^{\mathsf{max}},$$
$$s^{\mathsf{m}}_{k,\lambda}, s^{\mathsf{f}}_{k,\lambda} \in \{0,1\},$$

where $s_{k,\lambda} = \{s^{\mathsf{f}}_{k,\lambda}, s^{\mathsf{m}}_{k,\lambda}\}$. The above transformed problem is an integer linear programming (ILP) problem, which can be solved effectively by CVX. Let $\tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$ be the optimum of $(\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda$, then we can find the optimum of $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ as $\tilde{G}^{\mathsf{OSTS}\star}_{\mathcal{B},\eta} = \min_\lambda \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$. Moreover, it can be shown that when we increase λ, all $d_{k,\lambda}$ will decrease. Therefore, the maximum value of λ is λ^{max} satisfying $\mathcal{H}_0(\omega^{\mathsf{f}}_k, d_{k,\lambda^{\mathsf{max}}}) \geq f^{\mathsf{f,rq}}_k$, $\forall k \in \mathcal{B}$ and $\sum_{k\in\mathcal{B}} d_{k,\lambda^{\mathsf{max}}} \leq D^{\mathsf{max}}$. Note that we can stop the search process when there exists a λ such that $\tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda) \leq F^{\mathsf{f,max}}$. When the bisection search for η converges, we can find the optimum $\lambda^\star = \text{argmin}_\lambda \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$, and the optimal variables $s^{\mathsf{m}\star}_k = s^{\mathsf{m}}_{k,\lambda^\star}$, $s^{\mathsf{f}\star}_k = s^{\mathsf{f}}_{k,\lambda^\star}$, $s^{\mathsf{c}\star}_k = 1 - s^{\mathsf{m}\star}_k - s^{\mathsf{f}\star}_k$, $f^{\mathsf{f}\star}_k = s^{\mathsf{m}}_{k,\lambda^\star} f_{k,\lambda^\star} + s^{\mathsf{f}}_{k,\lambda^\star} f^{\mathsf{f,rq}}_k$, and $d^\star_k = s^{\mathsf{m}}_{k,\lambda^\star} d_{k,\lambda^\star} + (1 - s^{\mathsf{f}}_{k,\lambda^\star} - s^{\mathsf{m}}_{k,\lambda^\star}) d^{\mathsf{rq}}_k, \forall k \in \mathcal{B}$. The OSTS algorithm for feasibility verification of $(\mathcal{P}^{\mathsf{ext}}_{\mathcal{B}})$ is summarized in Algorithm 1.7.

b2) Iterative λ-update based two-stage algorithm (IUTS Alg.) This method can solve $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ with very low complexity via Lagrangian dual updates. Specifically, the dual function of $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ can be defined as $\mathcal{G}^{\mathsf{o}}(\lambda) = \min_{\tilde{\Omega}_4} \mathcal{L}(\tilde{\Omega}_4, \lambda)$, and the dual problem can be stated as

$$\max_\lambda \mathcal{G}^{\mathsf{o}}(\lambda) \quad \text{s. t.} \quad \lambda \geq 0. \tag{1.36}$$

Since the dual problem is always convex, $\mathcal{G}^{\mathsf{o}}(\lambda)$ can be maximized by using the standard sub-gradient method where the dual variable λ is iteratively updated as follows: $\lambda_n = \Bigg[\lambda_{n-1} + \delta_n \Big(\sum_{k\in\mathcal{B}} \left(s^{\mathsf{m}}_{k,\lambda_{n-1}} d_{k,\lambda_{n-1}} + s^{\mathsf{c}}_{k,\lambda_{n-1}} d^{\mathsf{rq}}_k \right) - D^{\mathsf{max}} \Big) \Bigg]^+$, where n denotes the iteration index, δ_n represents the step size, and $[a]^+$ is defined as $\max(0, a)$. The sub-gradient method is guaranteed to converge to the optimal value of λ for an initial primal point Ω_4 if the step size δ_n is chosen appropriately, e.g., $\delta_n \to 0$ when $n \to \infty$, which is met by setting $\delta_n = 1/\sqrt{n}$.

For a given λ_n, we can determine the primal variable $d_{k,\lambda_n} = \text{inv}(\mathcal{H}_2(\lambda_n))$. For given λ_n and d_{k,λ_n}, the primal problem becomes a linear program in $s_{k,\lambda_n}, \forall k \in \mathcal{B}$, which can be solved effectively by using standard linear optimization techniques. Moreover, the vertices in this problem

are the points where the $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s are either 0 or 1. Thus, *solving the relaxed problem will also return binary values* 0 *or* 1. However, once the $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s take values of 0 or 1, the decision on the application execution location (fog or cloud) may be trapped at a local optimal solution such that the required fog computing resources cannot be updated to improve the solution. To overcome this critical issue, the gradient projection method can be adopted to slowly update variables $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s as $\mathbf{s}_k^{(n+1)} = \mathbb{P}_{\Phi_k}\left(\mathbf{s}_k^{(n)} - \check{\delta}\nabla\mathbf{s}_k^{(n)}\right)$, where $\mathbf{s}_k^{(n)} = [s^{\mathsf{m}}_{k,\lambda_n}, s^{\mathsf{f}}_{k,\lambda_n}, s^{\mathsf{c}}_{k,\lambda_n}]$, $\check{\delta}$ is the step size, $\nabla\mathbf{s}_k^{(n)} = [\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_{k,\lambda_n}) + \lambda_n d_{k,\lambda_n}, \lambda_n f_k^{\mathsf{f,rq}}, \lambda_n d_k^{\mathsf{rq}}]$, and $\mathbb{P}_{\Phi_k}(.)$ is the projection onto the set $\Phi_k = \left\{\mathbf{s}_k | \mathbf{s}_k \geq 0,\ s^{\mathsf{f}}_{k,\lambda_n} + s^{\mathsf{c}}_{k,\lambda_n} + s^{\mathsf{m}}_{k,\lambda_n} \leq 1\right\}$. Finally, it can be verified that this iterative mechanism always converges [30].

1.2.2.4 Numerical Results

We consider a hierarchical fog-cloud system consisting of K=10 users where the users are randomly distributed in the cell coverage area with a radius of 800 m and the BS is located at the cell center. In particular, the path-loss is calculated as $\beta_k(\text{dB}) = 128.1 + 37.6\log_{10}(dist_k)$, where $dist_k$ is the geographical distance between user k and the BS (in km) [27]. We further set the beamforming gain as $M_0 = 5$, the maximum transmission bandwidth as $\rho_k^{\mathsf{max}} = 1$ MHz, and the noise power density as $\sigma_{\mathsf{bs}} = 1.381 \times 10^{-23} \times 290 \times 10^{0.9}$ W/Hz [31]. All users are assumed to have the same maximum clock speed of 2.4 GHz, a maximum transmit power of $p_k^{\mathsf{max}} = 0.22$ W, and the circuit power consumption per Hz is set to $p_{k,0} = 22$ nW/Hz. We assume that the number of transmission bits incurred to support computation offloading b_k^{in} is the same for all users. Moreover, the computation demands of the 10 users $\{c_1, c_2, ..., c_9, c_{10}\}$ are set randomly in the range $1.8 - 2.4$ Gcycles while the maximum delay time is to $T_k^{\mathsf{max}} = 1$ second, the non-offloadable load is $c_{k,0} = 0.1c_k$, and the offloadable load is $c_{k,1} = 0.9c_k$ for all users. We also set the energy coefficient as $\alpha_k = 0.1 \times 10^{-27}$ and the computing time at the cloud server as $T^{\mathsf{c}} = T_k^{\mathsf{max}}/5$. For the data compression algorithm, we set $\gamma_{k,1}^{\mathsf{co}} = 0.03 \times 2.6^{32.28}$, $\gamma_{k,2}^{\mathsf{co}} = 32.28$, $\gamma_{k,3}^{\mathsf{co}} = 0.3$, $\gamma_{k,1}^{\mathsf{de}} = 0.115$, $\gamma_{k,2}^{\mathsf{de}} = -0.9179$, $\gamma_{k,3}^{\mathsf{de}} = 0.046, \forall k$, $\omega_k^{\mathsf{u,min}} = 2.3$, and $\omega_k^{\mathsf{u,max}} = 2.9$. The energy and delay weights are chosen so that $w_k^{\mathsf{E}} + w_k^{\mathsf{T}} = 1, \forall k$. Simulation results are obtained by averaging over 100 realizations of the random locations of the users. Finally, for all figures, we set the raw data size as $b_k^{\mathsf{in}} = 4$ Mbits (except for Fig. 1.3), $w_k^{\mathsf{E}} = 2w_k^{\mathsf{T}}, \forall k$, the maximum fog computing resource as $F^{\mathsf{f,max}} = 15$ GHz, the maximum backhaul capacity as $D^{\mathsf{max}} = 20$ Mbps, and $\kappa = 50$ (except for Figs. 1.3), where κ captures the relationship between $\gamma_{k,0}^{\mathsf{u}}$ in (1.20) and the raw data size as $\gamma_{k,0}^{\mathsf{u}} = \kappa b_k^{\mathsf{in}}$ [32].

In practice, a fog server can support more powerful data compression algorithms compared to the users. This implies that the compression ratio for the fog server is much larger than that for the users. Therefore, when the fog server decompresses and re-compresses data, we set the parameters as follows: $\gamma_{k,1}^{\mathsf{co,f}} = 0.076, \gamma_{k,2}^{\mathsf{co,f}} = 0.7116, \gamma_{k,3}^{\mathsf{co,f}} = 0.5794$, $\omega_k^{\mathsf{f,min}} = 3.4$, and $\omega_k^{\mathsf{f,max}} = 11.2$. The step size is set as $\check{\delta} = 0.1$.

In Fig. 1.3, we show the significant benefits of data compression for computation offloading where the min-max WEDC (called WEDC for brevity) vs. b_k^{in} is plotted for six different schemes: the 'Local-execution' scheme in which all users' applications are executed locally; the 'Alg. in [15] (w/o Comp)' scheme in which the benchmark algorithm in [15] is applied with $\omega_k^{\mathsf{u}} = 1, \forall k$, and no data compression [4]; the 'JCORA Alg. w/o Comp' in which the proposed JCORA algorithm is applied with $\omega_k^{\mathsf{u}} = 1, \forall k$, and no data compression (the other variables are optimized as in the JCORA algorithm); and three other instances of the proposed JCORA algorithm with data compression and three different values of $\kappa = 50, 100, 200$ ($\kappa = \gamma_{k,0}^{\mathsf{u}}/b_k^{\mathsf{in}}$). To guarantee a fair comparison between the 'Alg. in [15] (w/o Comp)' scheme and our proposed schemes, we also apply MIMO and optimize the offloading decision and the allocation of the fog computing resources, transmit power, bandwidth, and local CPU clock speed for the 'Alg. in [15] (w/o Comp)' scheme. In addition, for the remaining variable d_k, we allocate the backhaul capacity equally to the users that offload their tasks to the cloud server.

As can be observed from Fig. 1.3, computation offloading can greatly improve the WEDC when there are sufficient radio and computing resources to support the offloading (e.g., the incurred amount of data is not too large). Specifically, computation offloading even without data compressioncan result in a significant reduction of the WEDC compared to local execution, especially when the incurred amount of data b_k^{in} is small such that the constrained radio resources do not limit performance. Furthermore, even without exploiting data compression, our proposed algorithm (JCORA Alg. w/o Comp) results in a much better performance than the algorithm proposed in [15]. This is because our proposed design jointly optimizes the offloading decisions and the computing and radio resource allocation, while in [15], the offloading decisions are found nearly independent of the computing and radio resource allocation. In particular, the semidefinite relaxation technique employed in [15] may not always guarantee the rank-1 condition for the optimized matrix. Joint optimiza-

[4] As discussed in Section I, this paper provides the first study of joint data compression and computation offloading in hierarchical fog-cloud systems. Therefore, the recent work [15] on computation offloading in hierarchical fog-cloud systems, which does not exploit DC, is selected as benchmark for performance comparison.

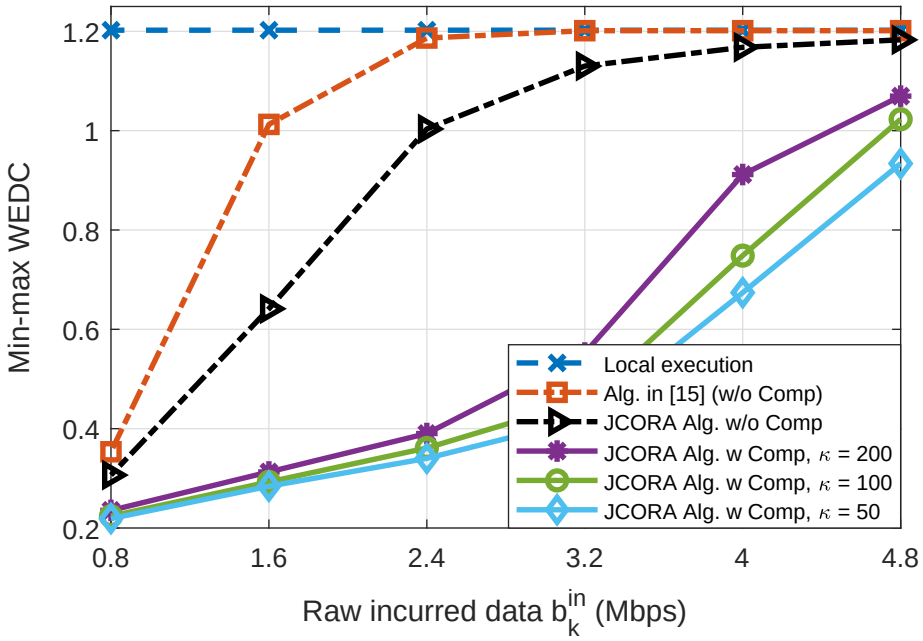

Figure 1.3 – Min-max WEDC vs. b_k^{in}.

tion of data compression, computation offloading, and resource allocation can lead to a significant further reduction of the WEDC for a larger range of b_k^{in} (e.g., when $b_k^{\mathsf{in}} = 2.4$ Mbps, the min-max WEDC is reduced by up to 65%). However, the energy and time consumed for (de)compression also affect the achievable min-max WEDC, and their impact tends to become stronger for larger $\gamma_{k,0}^{\mathsf{u}}$ and when the available radio resource is more limited.

To evaluate the system performance when data compression is performed at both the mobile users and the fog server, we consider the following parameter setting: $\gamma_{k,0}^{\mathsf{f}} = \gamma_{k,0}^{\mathsf{u}}$, $F^{\mathsf{f,max}} = 15$ GHz, and $D^{\mathsf{max}} = 20$ Mbps. The benefits of data re-compression at the fog are shown in Fig. 1.4 where we plot the min-max WEDC vs. b_k^{in} for four different schemes: the 'JCORA Alg. w Comp' scheme in which data are compressed only at the users while the three remaining schemes correspond to the proposed algorithms for the extended case. In particular, '9-pt PLA Alg. w Fog Comp', 'OSTS Alg. w Fog Comp', and 'IUTS Alg. w Fog Comp' correspond to the 9-point PLA, OSTS, and IUTS algorithms, respectively, which perform compression at both the users and the fog server. For $b_k^{\mathsf{in}} = 4$ Mbits, an additional min-max WEDC reduction of 35% can be achieved by performing data compression at both the users and the fog server. Moreover, the required radio resources decrease with decreasing b_k^{in}; therefore, the gain is reduced due to the decreasing demand for data transmission. When b_k^{in} increases, the main bottleneck for computation offloading are the limited radio resources available to support data transmissions between the users and the fog server; therefore, the gain

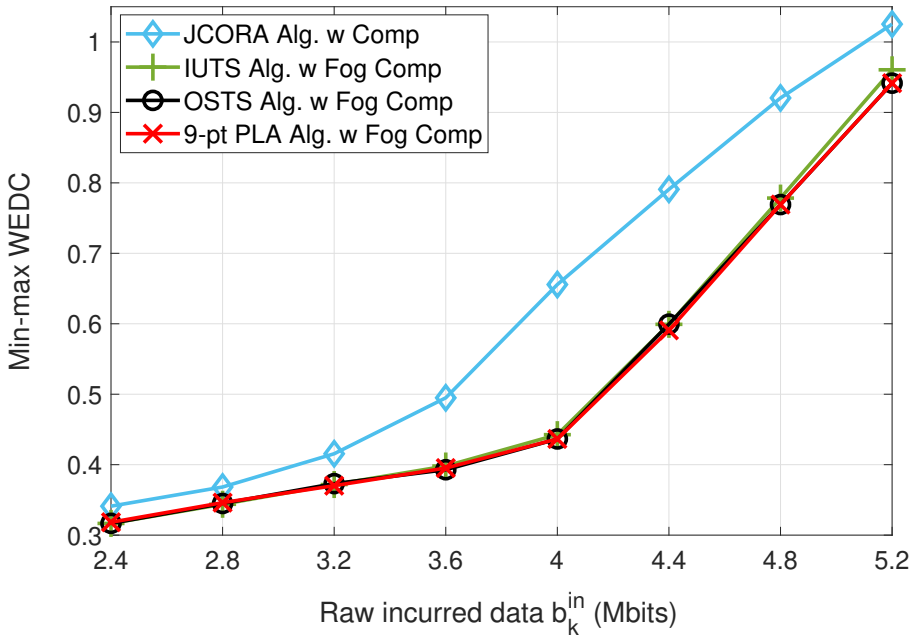

Figure 1.4 – Min-max WEDC in general design scenario.

due to data re-compression at the fog server becomes less significant. This figure also confirms that the '9-pt PLA', 'OSTS', and 'IUTS' schemes achieve almost the same min-max WEDC.

1.2.3 Wireless Scheduling for Heterogeneous Services with Mixed Numerology in 5G Wireless Networks

The main contributions of our work can be summarized as follows:

- 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.
- We propose two algorithms, named Resource Partitioning-based Algorithm (RPA) and Iterative Greedy Algorithm (IGA), to acquire efficient resource scheduling solutions.

1.2.3.1 System Model

We consider the 5G system where the available time-frequency resource is divided into resource elements (RE). Each RE occupies the bandwidth of Δ^{f}_{min} (Hz) and the slot duration of Δ^{t}_{min} (seconds).

The link/channel conditions for each subcarrier are assumed unchanged during the scheduling time. Moreover, we assume that the 2D RA is performed over each scheduling interval of $T = N^{\mathsf{t}}\Delta^{\mathsf{t}}_{\mathsf{min}}$ (seconds) and the bandwidth of $B^{\mathsf{f}} = M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}$ (Hz). Considering the scheduling problem for users where the serving base station supports multiple numerologies. The bandwidth of a PRB in numerology l is defined as Δ^{f}_l and the slot duration of a PRB in numerology l is defined as Δ^{t}_l. Then, we have $\Delta^{\mathsf{t}}_l = \Delta^{\mathsf{t}}_{l-1}/2$, $\Delta^{\mathsf{f}}_l = 2\Delta^{\mathsf{f}}_{l-1}$, $\Delta^{\mathsf{t}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{t}}_l, \forall l\}$, and $\Delta^{\mathsf{f}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{f}}_l, \forall l\}$. For convenience, the numerology used by user k is denoted as l_k, the set of all users is denoted as $\mathcal{K}$, the set of numerologies is denoted as $\mathcal{L}$, $l_{\mathsf{max}} = \max\{l \in \mathcal{L}\}$, $l_{\mathsf{min}} = \min\{l \in \mathcal{L}\}$, and $L = l_{\mathsf{max}} - l_{\mathsf{min}}$, and the cardinals of set $\mathcal{L}$ is denoted as $|\mathcal{L}|$. Each user k requires a data chunk of d^{rq}_k bits be completely transmitted and the total waiting time for its data transmission must not be larger than τ^{max}_k. We use (i, j) to refer to a particular RE where its location is given as $f \in [(i-1)\Delta^{\mathsf{f}}_{\mathsf{min}} : i\Delta^{\mathsf{f}}_{\mathsf{min}})$ and $t \in [(j-1)\Delta^{\mathsf{t}}_{\mathsf{min}} : j\Delta^{\mathsf{t}}_{\mathsf{min}})$, for $1 \le i \le M^{\mathsf{f}}$ and $1 \le j \le N^{\mathsf{t}}$.

1.2.3.2 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:

$$\boldsymbol{q}_{l,m,n} = \{(i,j) | m \le i \le m + M_l, n \le j \le n + N_l\}, \tag{1.37}$$

where $M_l = 2^{l-l_{\mathsf{min}}} - 1$, $N_l = 2^{l_{\mathsf{max}}-l} - 1$. 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 $x^{k,c}_{i,j}$'s, $y^{k,c}_{m,n}$'s where $y^{k,c}_{m,n} = 1$ if $\boldsymbol{q}_{l_k,m,n}$ corresponds to the c^{th} assigned PRB of user k, and $y^{k,c}_{m,n} = 0$ otherwise; $x^{k,c}_{i,j} = 1$ if RE (i, j) is assigned for user k in its c^{th} PRB, and $x^{k,c}_{i,j} = 0$ otherwise. For $k \in \mathcal{K}$, the ranges of c, i, j, m, and n are $c = 1 : C_k$, $i = 1 : M^{\mathsf{f}}$, $j = 1 : N^{\mathsf{t}}$, $m = 1 : M^{\mathsf{f}} - M_{l_k}$, and $n = 1 : N^{\mathsf{t}} - N_{l_k}$, respectively. We impose the following constraints to ensure non-overlapping RA:

$$\sum_{i'=m:m+M_{l_k}} \sum_{j'=n:n+N_{l_k}} x^{k,c}_{i',j'} \ge 2^L y^{k,c}_{m,n}, \ \forall k, c, m, n, \tag{1.38a}$$

$$\sum\nolimits_{k\in\mathcal{K}} \sum\nolimits_c x^{k,c}_{i,j} \le 1, \forall i, j, \text{ and } \sum\nolimits_m \sum\nolimits_n y^{k,c}_{m,n} \le 1, \forall k, c. \tag{1.38b}$$

Let $r_{m,n}^{k}$ denote the transmission rate of user k on PRB $\boldsymbol{q}_{l_k,m,n}$. Then, the total amount of data transmitted by user k during the scheduling interval is $d_k = \Delta_{l_k}^{\mathsf{t}} \sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}} \sum_{n=1}^{N^{\mathsf{t}}-N_{l_k}} \sum_{c=1}^{C_k} r_{m,n}^{k} y_{m,n}^{k,c}$.

Each user k wants its data chunk to be completely transmitted and the total waiting time be not larger than τ_k^{max}. Let $\tau_{k,0}$ denote the initial waiting time (of the data chunk) of user $k \in \mathcal{K}$ at the beginning of the considered scheduling interval.[5] 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 $\tau_{k,1} = \Delta_{\mathsf{min}}^{\mathsf{t}} \min\{j-1 \mid x_{i,j}^{k,c}=1, \forall i,j,c\}$.

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^{rq} bits with the total waiting time not larger than τ_k^{max}. To maintain these constraints, we define a function capturing if both constraints are satisfied as $u_k = \mathbb{1}_{d_k - d_k^{\mathsf{rq}}} \mathbb{1}_{\tau_k^{\mathsf{max}} - \tau_k}$, 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 $d_k \geq d_k^{\mathsf{rq}}$ and $\tau_k \leq \tau_k^{\mathsf{max}}$, respectively, we have $u_k = 1$; otherwise, $u_k = 0$. Then, the scheduling problem can be formulated as

$$(\mathcal{P}_1) \max_{\boldsymbol{x},\boldsymbol{y}} \sum\nolimits_{k\in\mathcal{K}} u_k \text{ s.t. } (1.38a), (1.38b), \text{ and } \boldsymbol{x}, \boldsymbol{y} \in \{0,1\},$$

where $\boldsymbol{x} = \{x_{i,j}^{k,c} | \, \forall i,j,k,c\}$ and $\boldsymbol{y} = \{y_{m,n}^{k,c} | \, \forall k,c,m,n\}$.

Let $z_{m,n}^{k,c} = u_k y_{m,n}^{k,c}, \forall k,c,m,n$, $(\mathcal{P}_1)$ can be transformed into the ILP form as

$$\begin{aligned}
(\mathcal{P}_1^{\mathsf{ILP}}) \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \sum_{k\in\mathcal{K}} u_k \\
& \text{s. t.} \quad (1.38a),\ (1.38b), \\
& \sum_{m} \sum_{n'=1:N_k^{\mathsf{rq}}} \sum_{c} z_{m,n'}^{k,c} - u_k \geq 0,\ \forall k \in \mathcal{K}, && \text{(1.39a)} \\
& \Delta_{l_k}^{\mathsf{t}} \sum_{m} \sum_{n} \sum_{c} r_{m,n}^{k} z_{m,n}^{k,c} - u_k d_k^{\mathsf{rq}} \geq 0,\ \forall k \in \mathcal{K}, && \text{(1.39b)} \\
& z_{m,n}^{k,c} \geq u_k + y_{m,n}^{k,c} - 1,\ \ z_{m,n}^{k,c} \leq \min\{u_k, y_{m,n}^{k,c}\},\ \forall k,c,m,n, && \text{(1.39c)}
\end{aligned}$$

[5] This initial waiting time is applied to the first chunk of a new data flow when the data flow arrives in the middle of the previous scheduling interval.

Algorithm 1.8. Resource Partitioning based Algorithm (RPA)

1: Initialize: Set initial value for M_{B}.
2: Partition resources into M_{B} sub-bands and distribute users into these sub-bands as in **Section 1.2.3.3**.
3: **Step 1**: Solve $(\mathcal{P}^{\mathsf{ILP}}_{1,m_{\mathsf{B}}})$ to obtain $u_k^{\mathsf{S1}\star}$ for all m_{B}, $k \in \mathcal{K}_{m_{\mathsf{B}}}$.
4: **Step 2**: Solve $(\mathcal{P}_{m_{\mathsf{B}}})$ to create a contiguous region of unallocated resources between two consecutive sub-bands while still satisfying the requirements of admitted users, i.e., users with $u_k^{\mathsf{S1}\star} = 1$, $\forall k$.
5: **Step 3**: Solve $(\mathcal{P}_{\mathsf{RPA}})$ to assign unallocated resources to un-admitted users, i.e., users with $u_k^{\mathsf{S1}\star} = 0$, $\forall k$.

Algorithm 1.9. Iterative Greedy Algorithm (IGA)

1: Initialize: $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_k, c_k = 0, \mathcal{W}_{m,n} = 1, W = 10$.
2: **repeat**
3: Compute $\mathcal{U}_{m,n,k}$ find the largest value of $\mathcal{U}_{m_0,n_0,k_0}$, and perform the corresponding PRB allocation.
4: Update different parameters after the PRB allocation as $c_{k_0} = c_{k_0} + 1$, assign $y^{k_0,c_{k_0}}_{m_0,n_0} = 1$, update the remaining required data $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_{k,0} - r^{k_0}_{m_0,n_0}\Delta^{\mathsf{t}}_{l_k}$, and $\mathcal{W}_{m_0,n} = W, \forall n = 1 : N^{\mathsf{t}}$.
5: Drop all overlapped PRBs $\boldsymbol{q}_{l_k,m,n}$ to PRB $\boldsymbol{q}_{l_{k_0},m_0,n_0}$
6: **until** $\mathcal{U}_{m,n,k} = 0, \forall m, n, k$

where $\boldsymbol{z} = \{z^{k,c}_{m,n}, z^{k,c}_{m,n} | \ \forall k, c, m, n\}$ and $\boldsymbol{u} = \{u^{\mathsf{r}}_k, u^{\mathsf{d}}_k | \ \forall k\}$. Here, N^{rq}_k represents the maximum number of time slots (with size of $\Delta^{\mathsf{t}}_{\mathsf{min}}$ seconds) that user k can wait, counting from the beginning of the scheduling interval, to meet its delay constraint which is determined as $N^{\mathsf{rq}}_k = \lfloor(\tau^{\mathsf{max}}_k - \tau_{k_0})/\Delta^{\mathsf{t}}_{\mathsf{min}}\rfloor$, where $\lfloor . \rfloor$ is the floor operation.

1.2.3.3 Proposed Algorithms

a) Resource Partitioning Based Algorithm (RPA)

We propose a low-complexity algorithm which solves $(\mathcal{P}^{\mathsf{ILP}}_1)$ by decomposing it into parallel small-scale sub-problems. We first divide the available bandwidth into M_{B} sub-bands where sub-band m_{B} occupies the spectrum from $(m_{\mathsf{B}} - 1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}}$ to $\min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}}, M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}\}$. Then, the following three steps are taken in RPA: 1) perform RA on each sub-band, 2) re-arrange unallocated resources for consecutive sub-bands, and 3) assign these re-arranged resources to support more (un-admitted) users.

The key steps of RPA are summarized in **Algorithm** 1.8. 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_{B} as $\mathcal{K}_{m_{\mathsf{B}}}$, and the set of REs in the frequency dimension as $\mathcal{I}_{m_{\mathsf{B}}} = \{m | m = (m_{\mathsf{B}}-1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor : \min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor, M^{\mathsf{f}}\}\}$. We then solve $(\mathcal{P}_1^{\mathsf{ILP}})$ corresponding to each sub-band m_{B} and the set of users $\mathcal{K}_{m_{\mathsf{B}}}$ to obtain admission decisions, denoted as $u_k^{\mathsf{S1}\star}$'s. The sub-problem for sub-band m_{B} is named $(\mathcal{P}_{1,m_{\mathsf{B}}}^{\mathsf{ILP}})$. In **Step 2**, after finding $u_k^{\mathsf{S1}\star}$'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 (1.37). The re-arrangement of unallocated REs on sub-band m_{B} can be achieved by solving the following problem:

$$\begin{aligned}(\mathcal{P}_{m_{\mathsf{B}}}) \quad & \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in\{0,1\}} \sum_{k\in\mathcal{K}_{m_{\mathsf{B}}}} \sum_{i\in\mathcal{I}_{m_{\mathsf{B}}}} \sum_{j=1:N^{\mathsf{t}}-N_{l_k}} \sum_{c=1:C_k} W_i x_{i,j}^{k,c} \\ \text{s.t.} \quad & u_k = u_k^{\mathsf{S1}\star}, \quad (1.38a), (1.38b), (1.39a), (1.39b), \ \forall k\in\mathcal{K}_{m_{\mathsf{B}}}, i\in\mathcal{I}_{m_{\mathsf{B}}},\end{aligned}$$

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 $(\mathcal{P}_{m_{\mathsf{B}}})$, 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 $\{\boldsymbol{x}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}, \boldsymbol{y}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}, \boldsymbol{z}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}\}$ denote the optimal solution of $(\mathcal{P}_{m_{\mathsf{B}}})$. In **Step 3**, we assign the unallocated resources, $\Omega = \{(i,j) | \boldsymbol{x}^\star_{\mathcal{P}_{m_{\mathsf{B}}}} = 0, \forall m_{\mathsf{B}}\}$, to the set of unadmitted users $\bar{\mathcal{K}} = \{k | u_k^{\mathsf{S1}\star} = 0\}$ by solving the following problem $(\mathcal{P}_{\mathsf{RPA}})$:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \sum_{k\in\bar{\mathcal{K}}} u_k \text{ s.t. } (1.38a), (1.38b), (1.39a), (1.39b), \forall k\in\bar{\mathcal{K}}, (i,j)\in\Omega.$$

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: $\mathcal{U}^k_{m,n} = \frac{r^k_{m,n}\Delta^{\mathsf{t}}_{l_k}}{d^{\mathsf{rq}}_{k,0}} \frac{n}{N^{\mathsf{rq}}_k} \mathbb{1}_{n\in\mathcal{N}_k} \mathcal{W}_{m,n}$ if $c_k \leq C_k, d^{\mathsf{rq}}_{k,0} > 0$, and $\mathcal{U}^k_{m,n} = 0$, otherwise, where $d^{\mathsf{rq}}_{k,0}$ is the remaining required data amount in each iteration, which is equal to d^{rq}_k in the first iteration, c_k is the current total PRBs assigned to user k, and $\mathcal{N}_k$ is the set of REs in the time domain, which is defined as $\mathcal{N}_k = \{n | n \leq N^{\mathsf{rq}}_k\}$ if $\sum_{n=1}^{N^{\mathsf{rq}}_k} \sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}} \sum_{c=1}^{C_k} y^{k,c}_{m,n} = 0$, and $\mathcal{N}_k = \{n | n \leq N^{\mathsf{t}}\}$, otherwise, and $\mathcal{W}_{m,n}$ 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 $\mathcal{W}_{m,n}$. In particular, $\mathcal{U}^k_{m,n}$ 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 1.9**. In each iteration of IGA, we need to compute the resource assignment weights $\mathcal{U}_{m,n,k}$ for all available m, n, k, determine the largest $\mathcal{U}_{m_0,n_0,k_0}$ to perform RA, update different parameters, and drop all overlapped PRBs to the assigned block. The worst-case complexity of each iteration is $\mathcal{O}(M^{\mathsf{f}}N^{\mathsf{t}}K)$. Let N^{iter} be the number of iterations, which is upper bounded by $M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|$, the overall worst-case complexity of IGA is $\mathcal{O}(N^{\mathsf{iter}}M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|K)$.

1.2.3.4 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 β_k (dB) $= 128.1 + 37.6 \log_{10}(\gamma_k)$ where γ_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 $\lfloor K/3 \rfloor$, $\lfloor K/3 \rfloor$ and $K - 2\lfloor K/3 \rfloor$, 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 2.8×10^5. The required data chunks d^{rq}_k over the interval T of 1 ms for users in groups A, B, and C are set

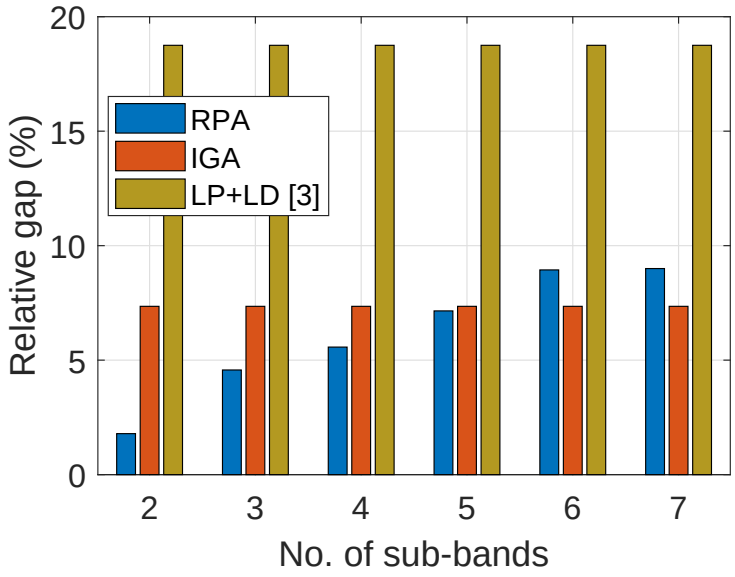

Figure 1.5 – Comparison of RPA and IGA with the optimum on the relative gap

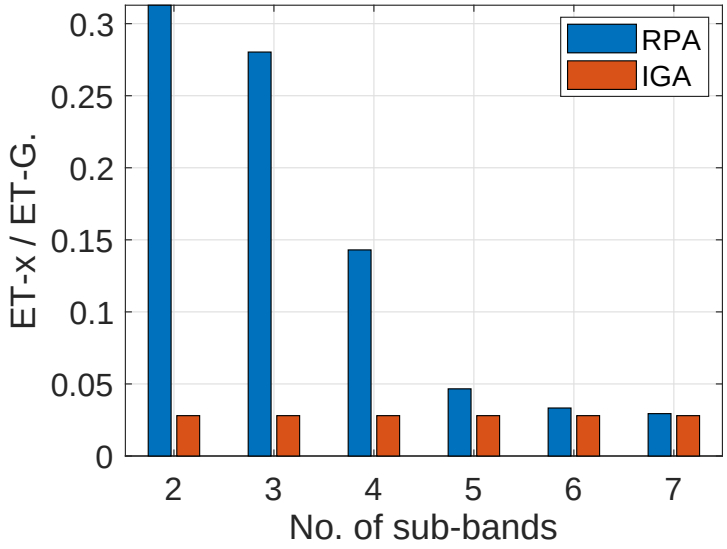

Figure 1.6 – Comparison of RPA and IGA with the optimum on the execution time

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 obtained by averaging the results over 50 random realizations.

We show the relative gap in Fig. 1.5 which is calculated as $(\sum_k u_k^{\mathsf{G}} - \sum_k u_k^{\mathsf{RPA/IGA}}) \times 100\% / \sum_k u_k^{\mathsf{G}}$ for $M^{\mathsf{f}} = 32$ and $K = 20$ where $u_k^{\mathsf{RPA/IGA/LP+LD[3]}}$ and u_k^{G} represent the objective values for user k obtained by using RPA/IGA/LP+LD[4] and the CVX-Gurobi solver, respectively. Fig. 1.6 shows the ratio between the average execution time (ET) of the RPA/IGA and that required to solve $(\mathcal{P}_1^{\mathsf{ILP}})$

by the CVX-Gurobi solver (ET-G). We have chosen u_k^{G} as a referenced solution because the relative gap between the actual optimal solution and the solution returned by the Gurobi solver is less than 10^{-4} by default. The "LP+LD" algorithm proposed in [4] 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. Specifically, the utility matrix is determined by considering the linear programming (LP) relaxation and the Lagrangian dual (LD) problem. In fact, the "LP+LD" algorithm obtains its solution by tackling the ILP dual problem where the zero duality-gap is generally not guaranteed. 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}.

1.3 Concluding Remarks

In this doctoral dissertation, we have developed various novel resource management techniques and algorithms for systems exploiting 5G NR and MEC. In particular, we have made three important research contributions. First, we have developed an energy-efficient resource allocation and offloading decision algorithms for MEC, which perform significantly better than the other conventional local computation strategies in terms of energy-saving and fairness. Second, we have proposed a general efficient weighted energy and delay cost for hierarchical fog-cloud systems via joint data compression, offloading decision, and resource allocation. The proposed designs significantly outperform the other state-of-the-art designs in the literature. Finally, 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 and demonstrate the efficacy of the proposed algorithms when determining the solutions.

Chapter 2

Résumé Long

2.1 Contexte et motivation

Les recherches sur le réseau cellulaire sans fil de cinquième génération (5G) et au-delà ont été motivées par la nécessité de prendre en charge l'explosion du trafic mobile et le nombre croissant de communications sans fil, y compris les connexions basées sur l'homme et l'internet des objets (IoT). Plus précisément, il est prévu que des dizaines de milliards d'appareils sans fil, de l'IoT à faible coût aux smartphones, tablettes, casques de réalité virtuelle, et voitures seront connectés aux réseaux sans fil au cours des prochaines années. La demande de communication sur différents types d'appareils mobiles dans des domaines verticaux, notamment l'usine intelligente, le véhicule intelligent, la grille intelligent, la ville intelligente, est de plus en plus sophistiquée. Ainsi, les futurs réseaux sans fil doivent fournir différents services de communication avec différentes exigences de QoS. En particulier, l'union internationale des télécommunications (ITU) classe les services de réseau mobile 5G en trois catégories: le mobile à large bande amélioré (eMBB), les communications de type de machines massives (mMTC) et les communications ultra-fiables et à faible latence (uRLLC). En général, eMBB fait référence aux services nécessitant une large bande passante, tels que la vidéo haute définition et les flux de réalité virtuelle/augmentée (VR/AR). En tant que tel, mMTC convient dans des scénarios avec une connectivité dense, tels que les villes intelligentes et l'agriculture intelligente. Ceci est en distinction avec uRLLC, qui est vise à prendre en charge les services de mise en réseau sensibles au facteur temps et les services critiques, tels que la conduite automatique/assistée et la télécommande. Pour fournir ces services, une nouvelle structure

de trame a été définie dans 5G New Radio (NR). En permettant la numérologie flexible, l'intervalle de temps de transmission (TTI) et la taille de trame peuvent être configurés de manière flexible pour s'adapter aux demandes de service [3]. Quelques premiers travaux [4–6] sur la planification et l'allocation des ressources avec la structure de trame 5G NR montrent une manière prometteuse d'améliorer les performances futures du système.

En général, les technologies de MEC/MCC permettent d'améliorer la convivialité mobile et de prolonger la durée de vie de la batterie mobile en déchargeant les applications gourmandes en calcul sur un serveur brouillard/cloud distant [7–9]. Dans un système MCC, d'énormes ressources de calcul sont disponibles dans le réseau principal, mais la capacité d'amenée limitée peut induire un retard important pour les applications sous-jacentes. En revanche, un système MEC, avec des ressources de calcul déployées à la périphérie du réseau à proximité des appareils mobiles, peut permettre le déchargement des calculs et répondre aux exigences exigeantes des applications [10]. Les systèmes informatiques de brouillard-nuage hiérarchiques qui tirent parti des avantages du MCC et du MEC peuvent améliorer les performances du système [11–15] où les serveurs de brouillard déployés à la périphérie du réseau peuvent fonctionner en collaboration avec les serveurs de nuage les plus puissants pour exécuter des applications utilisateur gourmandes en calculs. Plus précisément, lorsque les applications des utilisateurs nécessitent une puissance de calcul élevée ou une faible latence, leurs tâches de calcul peuvent être déchargées et traitées sur les serveurs de brouillard et/ou de cloud distants. Les scénarios potentiels et les applications de MEC et de ses variantes sont encore en discussion pour les systèmes 5G et au-delà.

En raison du besoin d'échange de données engendré par le processus de déchargement, la transmission sans fil joue un rôle intégral dans le système de déchargement informatique [16]. Par conséquent, pour utiliser efficacement la puissance de MEC, il faut développer des conceptions efficaces pour la gestion conjointe des ressources sans fil et informatiques. De plus, des technologies de communication avancées telles que le MIMO massif, le réseau hétérogène (HetNet) et le dispositif à dispositif (D2D), qui permettent d'améliorer l'efficacité spectrale, seront utilisées dans le MEC et ses variantes. En conséquence, la gestion conjointe de deux types de ressources différents devient très difficile. Plus précisément, à la différence des réseaux sans fil généraux, l'énergie et le temp délai dans MEC et ses variantes sont liés non seulement à la transmission sans fil, mais également aux facteurs de calcul. Ce sont les fonctions compliquées de différents paramètres et facteurs tels que la bande passante, la puissance de transmission, la vitesse d'horloge du processeur,

l'emplacement d'exécution. Une gestion et une optimisation efficaces de ces paramètres dans deux types de ressources différents sont un problème très difficile [17], et nécessitent encore beaucoup plus d'efforts concertés de la communauté sans fil [18, 19].

En outre, la vue de l'application est un aspect important lors de la mention d'un réseau sans fil 5G. En effet, avec les récentes percées dans l'intelligence artificielle (AI), de nouvelles applications émergentes permettant de nouvelles façons d'interactions entre les choses et les humains ont été créées pour améliorer la qualité de vie. Beaucoup d'entre elles sont des applications à forte intensité de calcul telles que la cybersanté, la reconnaissance/détection/surveillance d'objets. Lorsque seuls les problèmes liés aux communications sont concernés par la conception et la gestion du réseau, il est impossible d'activer ces applications exigeantes en calcul sur de nombreux types d'appareils, en particulier les appareils IoT à faible coût. Par conséquent, les réseaux sans fil 5G doivent prendre en charge non seulement les fonctions de communication, mais aussi de calcul, de contrôle et de distribution de contenu (4C). Le MEC a récemment été proposé comme une technologie importante dans les réseaux sans fil 5G pour permettre une variété de nouvelles applications à forte intensité de calcul, même sur des appareils IoT à faible coût. En général, MEC est un concept d'architecture de réseau défini par ETSI [20], qui permet des capacités de l'informatique en nuage et un environnement de service informatique à la périphérie du réseau cellulaire. Différents aspects de conception de MEC, tels que le partitionnement des tâches et l'allocation des ressources, ont été étudiés dans les communautés universitaires et industrielles pour les activer et prendre en charge les futurs scénarios et applications système [21, 22].

5G NR est une toute nouvelle interface radio en cours de développement pour la 5G afin de prendre en charge une grande variété de services et d'appareils. Cette partie présentera brièvement un nouveau concept en 5G NR, nommé numérologie. A noter que la numérologie est un terme qui est utilisé pour définir la grille de ressources discrètes dans le plan temps-fréquence continu. Une caractéristique critique du 5G NR est l'utilisation de la fréquence porteuse de moins de 1 GHz à 52,6 GHz, comme défini dans la version 15 du projet de partenariat de troisième génération (3GPP). Cependant, les propriétés de propagation des canaux des bandes de basses et hautes fréquences sont très différent. En particulier, la bande de basses fréquences est fortement affectée par les environnements intensifs à propagation de retard tandis que la bande de hautes fréquences est fortement influencée par le bruit de phase [23]. En conséquence, une numérologie unique appliquée à une large gamme de fréquences devient inefficace, voire impossible. Par rapport à la numérologie

LTE, le 5G NR prend en charge plusieurs types d'espacement de sous-porteuse sur la base d'un espacement de sous-porteuse de base de 15 kHz [24].

En particulier, 5G NR définit cinq numérologies OFDM distinctes qui sont paramétrées comme $\mu = 0, 1, 2, 3, 4$. La numérologie μ a la bande passante de sous-porteuse de $2^{\mu} \times 15$ kHz et une durée de tranche de $2^{-\mu}$ millisecondes. En numérologie $\mu = 0$, la grille temps-fréquence est la même que celle du LTE, 5G NR peut coexister avec LTE et NB-IoT basé sur LTE sur la même sous-porteuse. Pour les bandes de fréquences plus basses avec un espacement étroit entre les sous-porteuses, les numérologies 0, 1 et 2 sont utilisées pour contrer les environnements intensifs à propagation de retard. Pour les bandes de fréquences plus hautes, l'utilisation de la numérologie 2, 3 et 4 avec un large espacement des sous-porteuses peut rendre le système robuste au bruit de phase et peut bien prendre en charge les services à faible latence [24]. L'introduction de différentes numérologies offre la flexibilité pour divers services dans le même système. Elle également de nouveaux défis lors du multiplexage de différentes numérologies dans le même espace temps-fréquence. En raison de la différence de grille temps-fréquence, la planification sans fil pour des services hétérogènes avec une numérologie mixte dans les réseaux sans fil 5G devient un problème important pour améliorer les performances du système.

2.2 Contributions à la Recherche

La gestion des ressources de calcul et radio pour les systèmes MEC et la planification sans fil pour les services hétérogènes compte tenu de la numérologie mixte NR sont des problèmes de recherche pris en compte dans la thèse. A cette fin, nous considérons trois aspects de conception fondamentaux pour permettre la coexistence ci-dessus, à savoir l'allocation des ressources et la décision de déchargement dans les systèmes MEC, la compression conjointe des données, la décision de déchargement et l'allocation des ressources dans les systèmes informatiques de brouillard-nuage hiérarchiques et la planification sans fil pour les services hétérogènes avec numérologie mixte dans les réseaux sans fil 5G. Plus précisément, nos principales contributions peuvent être décrites comme suit.

2.2.1 Déchargement de Calcul dans les Systèmes MEC basés sur MIMO sous l'estimation de CSI parfaite et imparfaite

Dans cette contribution, nous développons des algorithmes pour optimiser le déchargement et l'allocation des ressources dans les systèmes MEC basés sur MIMO en considérant l'estimation de CSI parfaite/imparfaite. Les conceptions de déchargement de calcul existantes pour le scénario multi-tâches multi-utilisateurs n'ont pas pris en compte l'importante technologie de communication MIMO et ses problèmes connexes tels que l'estimation de CSI imparfaite. Notre article actuel vise à combler cette lacune dans la littérature en proposant des algorithmes généraux de déchargement et d'allocation des ressources qui peuvent assurer l'équité et prendre en compte la technologie de pointe MIMO. En particulier, les principales contributions de ce travail de recherche peuvent être résumées comme suit:

- Nous considérons deux scénarios importants avec une estimation de parfaite-CSI (P-CSI) et imparfaite-CSI (IP-CSI) pour le système MEC basé sur MIMO.
- Nous proposons différents algorithmes efficaces pour résoudre le problème de déchargement de calcul et d'allocation de ressources conjoint, qui minimise l'énergie consommée pondérée maximale (Min-max W.C.E) pour les utilisateurs mobiles compte tenu des contraintes de latence et de limitation des ressources. En particulier, nous proposons un algorithme optimal réalisant la solution optimale globale pour le scénario P-CSI, et deux algorithmes itératifs de faible complexité pour déterminer les solutions sous-optimales pour les scénarios P-CSI et IP-CSI, respectivement.
- Nous discutons de l'extension de la conception proposée lorsque le temps de renvoyer les résultats de calcul des utilisateurs déchargés est considéré. Nous décrivons également comment étendre l'algorithme proposé pour résoudre ce problème plus général.

2.2.1.1 Modèle de Système

Nous considérons un système MEC comprenant des K UEs à antenne unique et une BS, colocalisé avec le serveur de périphérie, équipé d'antennes M. Nous supposons que l'UE k a l'ensemble des tâches de calcul indépendantes $\mathcal{L}_k$, et chaque tâche $l_k \in \mathcal{L}_k$ nécessite un certain nombre de cycles

CPU c_{k,l_k} et un certain nombre de bits de données b_{k,l_k} (par exemple, pour transmettre les états de programmation impliqués à la BS). La décision de déchargement binaire pour la tâche $l_k \in \mathcal{L}_k$ est capturée par la variable binaire s_{k,l_k}, où $s_{k,l_k} = 1$ si la tâche l_k est exécuté sur l'appareil mobile, et $s_{k,l_k} = 0$ si cette tâche est déchargée sur le serveur de périphérie. Soit f_k et f_k^{c} la vitesse d'horloge du processeur pour exécuter l'application de UE k localement sur l'appareil mobile et à distance sur le serveur de périphérie, respectivement. Ensuite, pour l'utilisateur k, nous avons l'énergie de calcul locale $\xi_k^{\mathsf{lo}} = \alpha_k f_k^2 \sum\limits_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}$, le temps d'exécution local $t_k^{\mathsf{lo}} = f_k^{-1} \sum\limits_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}$, et le temps d'exécution à distance $t_k^{\mathsf{c}} = (f_k^{\mathsf{c}})^{-1} \sum\limits_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) c_{k,l_k}$, où α_k désigne le coefficient d'énergie spécifié dans le modèle de CPU.

Soit $\boldsymbol{h}_k \in \mathbb{C}^{M \times 1}$ le vecteur de gain de canal de liaison montante entre UE k et les antennes de la BS où les éléments du vecteur de gain de canal de liaison montante $\boldsymbol{h}_k$ sont modélisés comme $h_{mk} = \varphi_{mk}\sqrt{\beta_k}, m \in \{1, 2, ..., M\}$, où φ_{mk} et β_k représentent respectivement les coefficients d'évanouissement à petite et à grande échelle. Soit p_k la puissance d'émission de liaison montante de l'UE k et $\boldsymbol{n}$ le vecteur de bruit dont les composantes sont des i.i.d. $\mathcal{CN}(0, \sigma_{bs})$. L'ensemble des UE qui ne peuvent pas exécuter toutes leurs tâches localement soit noté $\mathcal{K}_\xi = \{k \in \mathcal{K} | \sum_{l \in \mathcal{L}_k} s_{k,l_k} < |\mathcal{L}_k|\}$ et la détection basée sur ZF est appliquée dans ce système considéré, la borne inférieure du débit de transmission moyen, la borne supérieure du temps de transmission moyen et de l'énergie de transmission moyenne de UE k pour le scénario P-CSI sont respectivement notées comme

$$r_k^{\mathsf{lb}} = W \log_2(1 + p_k \beta_k^{\mathsf{a}}), \tag{2.1}$$

$$t_{k,\mathsf{P}}^{\mathsf{t,ub}} = (r_k^{\mathsf{lb}})^{-1} \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}, \tag{2.2}$$

$$\xi_{k,\mathsf{P}}^{\mathsf{t,ub}} = (p_k + p_{k,0}) t_{k,\mathsf{P}}^{\mathsf{t,ub}}, \tag{2.3}$$

où $\beta_k^{\mathsf{a}} = \frac{(M - |\mathcal{K}_\xi|)\beta_k}{\sigma_{bs}}$.

Pour le scénario IP-CSI, soit T le nombre de périodes de symboles correspondant à l'intervalle de cohérence de canal et τ le nombre de symboles dans le pilote. Soit $\sqrt{\tau p^{\mathsf{tr}}}\boldsymbol{\phi}_k \in \mathbb{C}^{\tau \times 1}$ la séquence pilote affectée à UE k où p^{tr} désigne la puissance pilote et $||\boldsymbol{\phi}_k||^2 = 1$. En supposant que la détection

basée sur ZF est appliquée et que $\boldsymbol{\phi}_k^H \boldsymbol{\phi}_j = 0, \forall k \neq j$, la borne inférieure du débit de transmision moyen $\hat{r}_k^{\mathsf{lb}}$ est donné par [25]:

$$\hat{r}_k^{\mathsf{lb}} = W \log_2 \left(\boldsymbol{p}^{\mathsf{t}} \boldsymbol{\lambda}_k + \sigma_k + p_k \right) - W \log_2 \left(\boldsymbol{p}^{\mathsf{t}} \boldsymbol{\lambda}_k + \sigma_k \right), \tag{2.4}$$

où $\lambda_{k,i} = \frac{(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}) \sigma_{bs} \beta_i}{(\tau p^{\mathsf{tr}} \beta_i + \sigma_{bs}) \tau p^{\mathsf{tr}} \beta_k^2 (M - K_\xi)}$, $\sigma_k = \frac{(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}) \sigma_{bs}}{\tau p^{\mathsf{tr}} \beta_k^2 (M - K_\xi)}$. Ensuite, la borne supérieure du temps de transmission moyen et de l'énergie (y compris le temps d'entraînement et l'énergie) peuvent être écrites, respectivement, comme

$$t_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \frac{T}{T - \tau} t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{2.5}$$

$$\xi_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \left(\frac{\tau (p^{\mathsf{tr}} + p_{k,0})}{T - \tau} + p_k + p_{k,0} \right) t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{2.6}$$

où $t_{k,\mathsf{IP1}}^{\mathsf{t,ub}} = (\hat{r}_k^{\mathsf{lb}})^{-1} \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}$.

Le problème qui minimise l'énergie consommée pondérée maximale chez les utilisateurs mobiles compte tenu de la latence et des contraintes de ressources radio-calcul limitées peut être énoncé comme suit:

$$(\mathcal{P}_2) \quad \min_{\boldsymbol{S}, \boldsymbol{f}, \boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}, \xi} \quad \xi \tag{2.7a}$$

$$\text{s. t.} \quad w_k (\xi_k^{\mathsf{lo}} + \xi_k^{\mathsf{t}}) \leq \xi, \ \forall k, \tag{2.7b}$$

$$t_k^{\mathsf{lo}} \leq \eta_k, \quad \forall k, \tag{2.7c}$$

$$t_k^{\mathsf{t}} + t_k^{\mathsf{c}} \leq \eta_k, \ \forall k, \tag{2.7d}$$

$$s_{k,l_k} \in \{0, 1\}, \quad \forall k, \tag{2.7e}$$

$$\sum\nolimits_{k \in \mathcal{K}_\xi} f_k^{\mathsf{c}} \leq F^{\mathsf{c}}, \quad f_k^{\mathsf{c}} \geq 0, \tag{2.7f}$$

$$0 \leq f_k \leq F_k^{\mathsf{max}}, \quad \forall k, , \tag{2.7g}$$

$$0 \leq p_k \leq p_k^{\mathsf{max}}, \quad \forall k, \tag{2.7h}$$

où w_k représente le pondération énergétique de UE k, $\boldsymbol{S} = \{\boldsymbol{s}_k, \forall k\}, \boldsymbol{s}_k = \{s_{k,l_k}, \forall l_k\}$, $\{\boldsymbol{f}, \boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}\} = \{f_k, f_k^{\mathsf{c}}, p_k, \forall k\}$, η_k est le délai maximal permis de UE k, F_k^{max} indique la capacité de calcul maximale de UE k, et p_k^{max} représente la transmission maximale la puissance de l'UE k, F^{c} est le budget de

calcul disponible sur le serveur de périphérie, t_k^{t} et ξ_k^{t} peut être exprimé pour les scénarios P-CSI et IP-CSI comme

$$t_k^{\mathsf{t}}=\begin{cases} t_{k,\mathsf{P}}^{\mathsf{t,ub}},\ \text{P-CSI} \\ t_{k,\mathsf{IP}}^{\mathsf{t,ub}},\ \text{IP-CSI} \end{cases} \quad ; \quad \xi_k^{\mathsf{t}}=\begin{cases} \xi_{k,\mathsf{P}}^{\mathsf{t,ub}},\ \text{P-CSI} \\ \xi_{k,\mathsf{IP}}^{\mathsf{t,ub}},\ \text{IP-CSI} \end{cases} .$$

Proposition 2.1. *Le problème* $(\mathcal{P}_2)$ *peut être remanié comme*

$$\begin{aligned} (\mathcal{P}_2) \quad & \min_{\boldsymbol{S},\boldsymbol{f}^{\mathsf{c}},\boldsymbol{p},\xi} \quad \xi \\ & s.\ t. \quad \sum_{l_k\in\mathcal{L}_k} s_{k,l_k}c_{k,l_k} \le \eta_k F_k^{\mathsf{max}}, \quad \forall k\in\mathcal{K}, \qquad (2.8a) \\ & \qquad (2.7b),(2.7d)-(2.7f),(2.7h), \end{aligned}$$

où

$$\xi_k^{\mathsf{lo}} = \alpha_k \eta_k^{-2} \left(\sum_{l_k\in\mathcal{L}_k} s_{k,l_k}c_{k,l_k} \right)^3, \quad \forall k\in\mathcal{K}, \qquad (2.9)$$

2.2.1.2 Conception d'Algorithmes pour le Scénario P-CSI

Pour résoudre le difficile MINLP $(\mathcal{P}_2)$, nous proposons deux algorithmes où le premier (P-O) peut trouver la solution optimale globale tandis que le second (P-SO) réalise une solution avec une complexité moindre.

a) P-CSI - L'algorithme Optimal (P-O) :

Proposition 2.2. *Pour une valeur donnée de* ξ*, le problème* $(\mathcal{P}_2)$ *est réalisable si tous les sous-problèmes* $(\mathcal{P}_3)_k, \forall k$ *sont réalisables et* $\sum_{k\in\mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}} \le F^{\mathsf{c}}$ *où* $f_{k,\xi}^{\mathsf{c,min}}$ *est la valeur optimale de* $(\mathcal{P}_3)_k$*,* $\mathcal{K}_\xi$ *est le l'ensemble des UEs qui téléchargent leurs tâches de calcul si l'exécution locale consomme l'énergie totale supérieure à* ξ*, qui est calculée comme suit:*

$$|\mathcal{K}_\xi| = \sum_k \delta_k, \ \textit{et}\ \delta_k = \begin{cases} 1,\ \textit{si}\ w_k\xi_k^{\mathsf{lo}} > \xi, \forall l_k\ \textit{st}\ s_{k,l_k}=1 \\ 0,\ \textit{autrement} \end{cases} . \qquad (2.10)$$

Algorithm 2.1. Optimal Algorithm - P-CSI (P-O)

1: **Initialize**: choose ϵ, $\xi_{\min} = 0$ and $\xi_{\max} = \min(\max(w_k \xi_k^{\mathsf{lo}} | s_{k,l_k} = 1, \sum_{l_k \in \mathcal{L}_k} c_{k,l_k} \leq \eta_k F_k^{\mathsf{max}}), \xi^{\infty})$.

2: **while** $\xi_{\max} - \xi_{\min} < \epsilon$ **do**
3: Assign $\xi = (\xi_{\max} + \xi_{\min})/2$.
4: Determine set $\mathcal{K}_\xi$ as in (2.10).
5: Solve $(\mathcal{P}_3)_k$ to get $f_{k,\xi}^{\mathsf{c,min}}$ for all $k \in \mathcal{K}_\xi$.
6: Assign $feasibility = true$ if all subproblems $(\mathcal{P}_3)_k$ are feasible and $\sum_{k \in \mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}} \leq F^{\mathsf{c}}$.
7: Assign $(\xi_{\max}, \xi_{\min}) = \text{bisectionSearch}(feasibility, \xi)$
8: **end while**

et $(\mathcal{P}_3)_k$ *sont les sous-problèmes qui peuvent être résolus indépendamment par les UE individuels* $(k \in \mathcal{K}_\xi)$ *pour des valeurs données de* ξ *et* $|\mathcal{K}_\xi|$*, qui sont donnés comme*

$$(\mathcal{P}_3)_k \quad \min_{\boldsymbol{s}_k, f_k^{\mathsf{c}}, p_k} \quad [f_k^{\mathsf{c}}]^+$$
$$s.\ t. \quad (2.7b)_k, (2.7d)_k, (2.7e)_k, (2.7h)_k, (2.8a)_k,$$

où $[f_k^{\mathsf{c}}]^+ = \max(f_k^{\mathsf{c}}, 0)$*, les contraintes* $(2.7b)_k$*,* $(2.7d)_k$*,* $(2.7e)_k$*,* $(2.7h)_k$*, et* $(2.8a)_k$ *dénotent les contraintes correspondantes* $(2.7b)$*,* $(2.7d)$*,* $(2.7e)$*,* $(2.7h)$*, et* $(2.8a)$ *pour UE* k*, respectivement.*

En utilisant les résultats de la *Proposition 2.2*, nous proposons un algorithme optimal pour résoudre le problème $(\mathcal{P}_2)$ comme décrit dans Algorithme 2.1. De plus, pour $\boldsymbol{s}_k$ donné, $(\mathcal{P}_3)_k$ est transformé de manière équivalente en problème convexe standard comme indiqué dans la Section 5.4.1. Ensuite, en considérant toutes les différentes combinaisons de s_{k,l_k} $(l_k \in \mathcal{L}_k)$, où $\boldsymbol{s}_k$ satisfait $(2.8a)_k$ et $(2.7e)_k$, nous pouvons déterminer $f_{k,\xi}^{\mathsf{c,min}}$ ainsi que la solution de $(\mathcal{P}_3)_k$.

b) P-CSI - L'algorithme de faible complexité (P-SO)

Nous proposons un algorithme de faible complexité qui résout de manière itérative deux sous-problèmes décomposés à partir de $(\mathcal{P}_2)$ où le premier, à savoir le sous-problème d'optimisation du déchargement (OP), détermine la décision de déchargement et l'allocation des ressources de calcul tandis que le second, c'est-à-dire le sous-problème d'allocation de puissance (PA), effectue l'allocation de puissance en liaison montante et réaffecte la ressource de calcul. Tout d'abord, pour une valeur donnée de $\boldsymbol{p}$, le sous-problème (OP) est donné comme suit:

$$(\mathcal{P}_2^{\mathsf{OP}}) \min_{\boldsymbol{S}, \boldsymbol{f}^{\mathsf{c}}, \xi} \xi \quad \text{s.t} \quad (2.7b), (2.7d) - (2.7f), (2.8a).$$

Deuxièmement, avec la solution de déchargement $\boldsymbol{S}$ obtenue en résolvant $(\mathcal{P}_2^{\mathsf{OP}})$, le (PA) est donné comme

$$(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}) \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\xi} \xi \quad \text{s.t} \quad (2.7b),(2.7d),(2.7f),(2.7h).$$

L'algorithme proposé (P-SO), qui résout de manière itérative $(\mathcal{P}_2^{\mathsf{OP}})$ et $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ jusqu'à convergence, est décrit dans Algorithme 2.2. En outre, cette approche est la clé pour résoudre le problème $(\mathcal{P}_2)$ dans le scénario IP-CSI lorsque la recherche d'une solution optimale serait impossible. Notez que $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$ est l'optimal du sous-problème $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ à l'itération q. Nous décrivons comment résoudre (OP) et (PA) dans ce qui suit.

Afin de s'attaquer à $(\mathcal{P}_2^{\mathsf{OP}})$, nous appliquerons la technique de décomposition utilisée dans la section 2.2.1.2 pour décomposer davantage ce sous-problème en sous-problèmes à petite échelle des utilisateurs individuels:

$$(\mathcal{P}_2^{\mathsf{OP}})_k \quad \min_{\boldsymbol{s}_k, f_k^{\mathsf{c}}} \quad [f_k^{\mathsf{c}}]^+ \text{s.t.} \quad (2.7b)_k,(2.7d)_k,(2.7e)_k,(2.8a)_k.$$

Pour déterminer un minimum de f_k^{c} pour un ξ donné, noté $f_{k,\xi}^{\mathsf{c,min}}$, nous appliquons la recherche de bissection sur ξ comme dans l'algorithme 2.1, à l'exception d'une certaine différence à l'étape 4 et à l'étape 5 pour trouver $f_{k,\xi}^{\mathsf{c,min}}$. Soit $\boldsymbol{S}_k^{\mathsf{bi}} \in \mathbb{R}^{2^{|\mathcal{L}_k|}\times|\mathcal{L}_k|}$ désigne la matrice binaire dont les lignes représentent toutes les combinaisons possibles de décisions de déchargement de tâche de UE k. Par exemple, $\boldsymbol{S}_k^{\mathsf{bi}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}^{\mathsf{t}}$ dans le cas de $|\mathcal{L}_k| = 2$. Ensuite, il peut être vérifié que la valeur minimale de f_k^{c} pour une valeur donnée de ξ peut être calculée comme

$$\begin{aligned}
f_{k,\xi}^{\mathsf{c,min}} &= \min((LC0_k \odot LC2_k \odot L20_k)\backslash\{0\}), \\
LC0_k &= \mathbb{1}_{\xi\times\mathbf{1}_{2^{|\mathcal{L}_k|}} - LC0'_k \geq 0}, \\
LC0'_k &= w_k\alpha_k\eta_k^{-2}(\boldsymbol{S}_k^{\mathsf{bi}}\boldsymbol{c}_k)^3 + \frac{w_k(p_k+p_{k,0})(1-\boldsymbol{S}_k^{\mathsf{bi}})\boldsymbol{b}_k}{r_k^{\mathsf{lb}}}, \\
LC2_k &= \left[(1-\boldsymbol{S}_k^{\mathsf{bi}})\boldsymbol{c}_k \oslash (\eta_k\times\mathbf{1}_{2^{|\mathcal{L}_k|}} - \frac{(1-\boldsymbol{S}_k^{\mathsf{bi}})\boldsymbol{b}_k}{r_k^{\mathsf{lb}}})\right]^+, \\
L20_k &= \mathbb{1}_{\eta_k F_k - \boldsymbol{S}_k^{\mathsf{bi}}\boldsymbol{c}_k \geq 0},
\end{aligned} \tag{2.12}$$

Algorithm 2.2. Low-complexity Algorithm - P-CSI (P-SO)

1: **Initialize**: choose ϵ, initial $p_k^{(0)} = p_{\max}/2, \forall k$.
2: **while** $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})} - \xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{PA}})} < \epsilon$ **do**
3: Assign $q = q + 1$;
4: Solve $\mathcal{P}_2^{\mathsf{OP}}$ to get $\boldsymbol{S}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})}$.
5: Solve $\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}$ to get $\boldsymbol{p}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$.
6: **end while**

où $\odot$ et $\oslash$ désignent respectivement le produit et la division Hadamard, $\mathbf{1}_n$ représente le vecteur $n \times 1$ de uns, $\mathbb{1}_{\boldsymbol{x} \geq 0}$ est la fonction d'indicateur et $\boldsymbol{x}^+ = \max(\boldsymbol{x}, 0)$. Dans les expressions ci-dessus, les éléments de $LC0_k$ et $L20_k$ seront égaux à 1 si la ligne correspondante de $\boldsymbol{S}_k^{\mathsf{bi}}$ satisfait la contrainte $(2.7b)_k$ et $(2.8a)_k$, respectivement. Le vecteur de $LC2_k$ décrit la valeur minimale de f_k^{c} correspondant à chaque ligne de $\boldsymbol{S}_k^{\mathsf{bi}}$. Il est à noter que $LC0'_k, LC2_k$ et $L20_k$ ne dépendent pas de la valeur de ξ; ainsi, nous avons juste besoin de les calculer au début de la recherche de bissection (la 'boucle while' dans l'algorithme 2.1) et de les utiliser pour mettre à jour $LC0_k$ et $f_{k,\xi}^{\mathsf{c,min}}$ correspondant à la valeur mise à jour de ξ.

Avec la solution de $(\mathcal{P}_2^{\mathsf{OP}})$, nous pouvons alors résoudre le $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ sous-problème à obtenir les solutions optimales de puissance d'émission $\boldsymbol{p}_k$ et d'allocation des ressources informatiques $\boldsymbol{f}^{\mathsf{c}}$. Cela peut être accompli en utilisant un processus similaire employé dans la section 2.2.1.2 qui applique la recherche de bissection sur ξ et en résolvant le sous-problème $(\mathcal{P}_3)'_k$. Il est à noter que l'algorithme 2.2 crée une séquence de solutions réalisables pour $(\mathcal{P}_2)$ où la valeur de la fonction objectif de ce problème diminue de façon monotone au fil des itérations.

2.2.1.3 Conception d'Algorithmes pour le Scénario IP-CSI

Nous nous attaquons au problème $(\mathcal{P}_2)$ en résolvant itérativement deux sous-problèmes $(\mathcal{P}_2^{\mathsf{OP}})$et $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ jusqu'à convergence . Étant donné que l'IP-CSI n'affecte que l'énergie de transmission et le temps de transmission, le sous-problème OP $(\mathcal{P}_2^{\mathsf{OP}})$ peut être résolu comme dans la section 2.2.1.2, il suffit de considérer le sous-problème PA $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$, qui peut être écrit comme

$$
\begin{aligned}
(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}}) \quad \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}}} \quad & \xi \\
\text{s. t.} \quad & (2.7b): \; w_k(\xi^{\mathsf{t,ub}}_{k,\mathsf{IP}} + \xi^{\mathsf{lo}}_k) \leq \xi, \\
& (2.7d): \; t^{\mathsf{t,ub}}_{k,\mathsf{IP}} + \frac{c^{\mathsf{a}}_k}{f^{\mathsf{c}}_k} \leq \eta_k, \;\; (2.7f), (2.7h).
\end{aligned}
$$

On peut vérifier que $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ est non convexe et NP-difficile. Pour s'y attaquer, nous appliquons d'abord DC pour convexifier approximativement les contraintes non convexes (2.7b) et (2.7d) pour une valeur donnée de ξ comme suit

$$
\begin{aligned}
p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}} \tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \quad &\leq \quad 0, \qquad (2.13) \\
\left(\frac{T}{T-\tau}\right) \frac{b^{\mathsf{a}}_k}{\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c^{\mathsf{a}}_k}{f^{\mathsf{c}}_k} - \eta_k \quad &\leq \quad 0. \qquad (2.14)
\end{aligned}
$$

où

$$
\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}) = W \log_2(p_k + \boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - W \log_2((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - \frac{\boldsymbol{\lambda}_k(\boldsymbol{p} - \boldsymbol{p}^{(q)})}{\log(2)\Big((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\Big)}, \quad (2.15)
$$

et $\boldsymbol{p}^{(q)}$ est le point que $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}) = \hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$.

Il est à noter que $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq \hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$ du fait que $v_k(\boldsymbol{p}) \leq \tilde{v}_k(\boldsymbol{p}) = v_k(\boldsymbol{p}^{(q)}) + \nabla v_k(\boldsymbol{p}^{(q)})(\boldsymbol{p} - \boldsymbol{p}^{(q)})$ au point $\boldsymbol{p}^{(q)}$, où $v_k(\boldsymbol{p}) = W \log_2\Big(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\Big)$. A partir de (2.15), la vérification de faisabilité de $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ pour une valeur donnée de ξ équivaut désormais à trouver au moins un point ($\boldsymbol{f}^{\mathsf{c}}$, $\boldsymbol{p}$, $\boldsymbol{p}^{(q)}$) qui crée des contraintes (2.7f) et (2.7h) et des inégalités (2.13), (2.14) réalisable. À cette fin, nous mettrons à jour de manière itérative $\boldsymbol{p}^{(q)}$ pour affiner l'approximation dans (2.15) et trouver

Algorithm 2.3. PA Feasibility Verification - IP-CSI

Initialize: choose $\boldsymbol{p}^{(0)}$ as the previous solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$.
repeat
  $q = q + 1$;
  At $\boldsymbol{p} = \boldsymbol{p}^{(q-1)}$, solve $(\mathcal{P}^{\mathsf{PA}}_{2})^{(q-1)}$ to get $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$
  if $\chi < 0$ **then**
    Assign $feasibility = true$
    Return $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$; **break**;
  else
    Assign $feasibility = false$
    Compute $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ for all k
  end if
until convergence

le minimum χ pour un donné $\boldsymbol{p}^{(q)}$, où χ est l'objectif du problème suivant:

$$(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)} \quad \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\chi} \quad \chi \tag{2.16a}$$

$$\text{s. t.} \quad p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}} \tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq \chi, \tag{2.16b}$$

$$\left(\frac{T}{T-\tau}\right) \frac{b^{\mathsf{a}}_k}{\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c^{\mathsf{a}}_k}{f^{\mathsf{c}}_k} - \eta_k \leq \chi, \tag{2.16c}$$

$$(2.7f), (2.7h),$$

Comme indiqué ci-dessus, la convexité de $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)}$ est garantie; par conséquent, nous pouvons résoudre efficacement ce problème en utilisant le solveur CVX. Enfin, la vérification de faisabilité est présentée dans l'algorithme 2.3, et la recherche de bissection pour résoudre $(\mathcal{P}^{\mathsf{PA}}_{2,\mathrm{IP}})$ est similaire à l'algorithme 2.1, à l'exception de la différence à l'étape 5, où la vérification de faisabilité est effectuée comme décrit dans l'algorithme 2.3. On peut vérifier que pour un ξ donné, en utilisant DC pour approximer le débit de transmission (en utilisant la limite inférieure du débit dans (2.15)) et en résolvant de manière itérative le problème $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(q)}$ conduit à la convergence.

Proposition 2.3. *Si la valeur optimale de $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ est égale à zéro à la convergence de l'algorithme 2.3, alors la solution de $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ combinant avec ξ donne un point stationnaire du sous-problème $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$, où Q désigne l'indice d'itération final à la convergence de l'algorithme 2.3.*

2.2.1.4 Considération de Transmission en Liaison Descendante

Nous étendons la conception proposée décrite dans le précédent sections pour examiner cette transmission de liaison descendante. Soit b_{k,l_k}^{dl} le nombre de bits de liaison descendante liés au résultat du calcul de la tâche l_k, qui doivent être envoyés de la BS à l'UE k. En supposant que le précodeur ZF est utilisé, le débit de liaison descendante ergodique de la borne inférieure peut être exprimé comme

$$\hat{r}_k^{\mathsf{dl,lb}} = \begin{cases} W \log_2\left(1 + p_k^{\mathsf{dl}}/\sigma_k^{\mathsf{dl}}\right) & \text{(P-CSI)} \\ W \log_2\left(1 + \frac{p_k^{\mathsf{dl}}}{\sum\limits_{i \in \mathcal{K}_\xi} p_i^{\mathsf{dl}} \lambda_{k,i} + \sigma_k^{\mathsf{dl}}}\right) & \text{(IP-CSI)} \end{cases}, \tag{2.17}$$

$$\frac{T}{T-\tau}\left(\frac{b_k^{\mathsf{a}}}{\hat{r}_k^{\mathsf{lb}}} + \frac{b_k^{\mathsf{a,dl}}}{\hat{r}_k^{\mathsf{dl,lb}}}\right) + \frac{c_k^{\mathsf{a}}}{f_k^c} \leq \eta_k, \quad \forall k \in \mathcal{K}_\xi, \tag{2.18}$$

where $b_k^{\mathsf{a,dl}} = \sum\limits_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}^{\mathsf{dl}}$. D'autre part, la puissance d'émission moyenne de BS peut être calculée comme

$$p_{\mathsf{BS}} = \sum_{k \in \mathcal{K}_\xi} p_k^{\mathsf{dl}} \mathbb{E}(||\hat{\boldsymbol{a}}_{\boldsymbol{k}}||^{\mathbf{2}}) = \begin{cases} \sum\limits_{k \in \mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}}{(M - |\mathcal{K}|_\xi)\beta_k}, \text{ P-CSI} \\ \sum\limits_{k \in \mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs})}{(M - |\mathcal{K}|_\xi)\tau p^{\mathsf{tr}} \beta_k^2}, \text{ IP-CSI} \end{cases}.$$

La puissance de transmission totale à BS doit être limitée par sa puissance maximale $p_{\mathsf{max}}^{\mathsf{dl}}$, qui peut être exprimée comme suit:

$$p_{\mathsf{BS}} \leq p_{\mathsf{max}}^{\mathsf{dl}}. \tag{2.19}$$

Nous pouvons maintenant formuler le problème de déchargement de calcul et d'allocation de ressources conjoint en considérant les transmissions de données en liaison montante et en liaison descendante comme suit:

$$(\mathcal{P}_2^{\mathsf{ext}})\ \min \xi \quad \text{s.t} \quad (2.7b), (2.7e), (2.7f), (2.7h), (2.8a), (2.18), (2.19).$$

Pour résoudre ce problème difficile, nous pouvons à nouveau le décomposer en deux sous-problèmes comme dans les sections précédentes. En particulier, nous résolvons de façon itérative le sous-problème (OP) (avec des contraintes (2.7b), (2.7e), (2.7f), (2.8a), et (2.18)) pour trouver le $\boldsymbol{s}, \boldsymbol{f}^{\mathsf{c}}$ optimal et résoudre le problème étendu (PA) sous-problème (avec contraintes 2.7b), (2.7f), (2.7h), (2.18), et (2.19) pour trouver $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$. Étant donné que les variables d'optimisation $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$ ne sont capturées que dans le fichier sous-problème étendu (PA), nous pouvons résoudre le sous-problème (OP) comme dans la section 2.2.1.2. Pour résoudre le sous-problème étendu (PA), nous pouvons appliquer les mêmes techniques que dans la section 2.2.1.3 pour traiter le débit de liaison descendante, car les deux composantes de retard, qui correspondent au temps de transmission de liaison montante de les données encourues et le temps de transmission du téléchargement du résultat du calcul, respectivement, ont la même structure.

2.2.1.5 Résultats Numériques

Nous considérons un système MEC avec la bande passante du canal de 10 MHz et $K = 20$ UEs répartis de manière aléatoire dans une zone de couverture cellulaire avec un rayon de 900m. Dans nos paramètres de simulation, nous définissons $F_k^{\mathsf{max}} = 2.4$ GHz, $P_k^{\mathsf{m}} = 0.22$ (Watts), $p_{k,0} = 0.05$, $\alpha_k = 0.1{\times}10^{-27}$, $|\mathcal{L}_k| = 5$, and $\eta_k = \eta$ pour tous k, $M = 30$, $F^{\mathsf{c}} = 40$ GHz, $p_{\mathsf{max}}^{\mathsf{dl}} = 10$ (Watts), $\sigma_{bs} = \sigma_k^{\mathsf{dl}} =$ bande passante $\times 3.6 \times 10^{-21}$, $T = 200$ symboles. Tous les UE ont le même nombre de tâches parallèles et la même demande de calcul totale de 0.24 Gcycles, mais le nombre de cycles CPU par tâche est défini de manière aléatoire. Le nombre total de bits de transmission pour toutes les tâches est défini pour être le même pour tous les UE tandis que le nombre de bits par tâche est généré de manière aléatoire. Pour l'évaluation des performances de la conception proposée, nous choisissons le rapport entre le nombre total de bits de transmission et le nombre total de cycles CPU requis (BPC) à environ 4.2×10^{-3} (à l'exception des résultats de la Fig. 2.1), qui est proche de sa valeur la plus élevée possible pour les applications considérées dans [26]. Le coefficient du

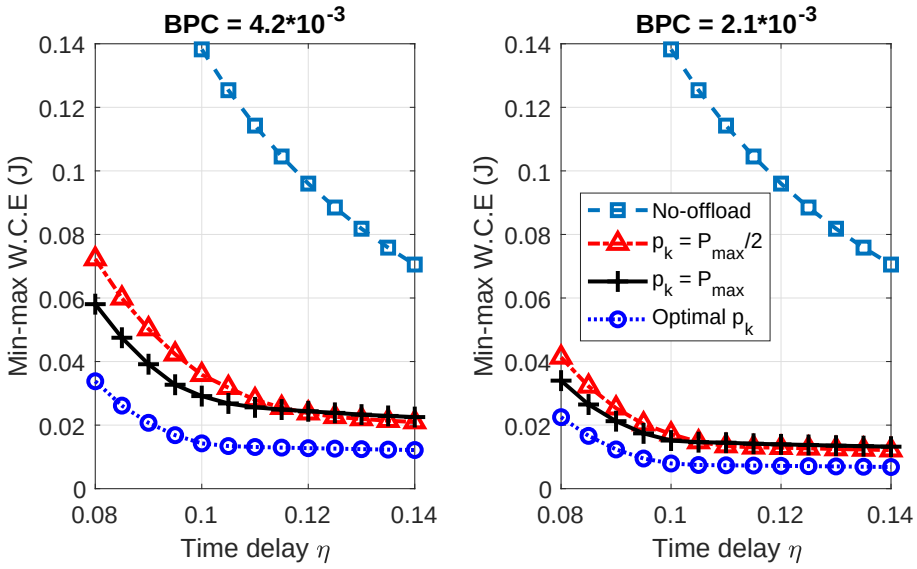

Figure 2.1 – Comparaison des performances avec / sans déchargement et avec/sans optimisation des ressources radio et calcul.

canal à petite échelle est généré selon la distribution de Rayleigh et l'affaiblissement de propagation est défini selon le rapport technique 3GPP comme β_k (dB) $= 128.1 + 37.6 \log_{10}(d_k)$ où d_k est la distance géographique entre UE k et la BS (en km) [27].

L'avantage de l'optimisation de l'allocation des ressources radio et calcul conjointe dans la conception de déchargement de calcul est illustré sur la Fig.2.1 pour faire varier le retard maximum autorisé η. Dans cette figure, compte tenu de l'absence de transmission de données en liaison descendante et du scénario P-CSI, nous comparons les performances réalisables dans quatre scénarios: traitement des tâches sur les appareils mobiles ('No-offload'), déchargement partiel avec décision de déchargement optimale et allocation des ressources en nuage avec fixe puissance d'émission pour tous les UE $p_k = p_{\max}/2$ et $p_k = p_{\max}$, et avec une allocation optimale de la puissance d'émission ('Optimal p_k '). Les sous-figures gauche et droite montrent le W.C.E min-max atteint pour différentes valeurs de bits de transmission par cycle CPU (BPC). Sur cette figure, nous pouvons voir que la latence minimale requise pour que l'appareil mobile puisse traiter ses tâches localement (le cas 'No-offload') est de 0,1 s tandis que la latence minimale requise dans les autres cas, 0,08 s. Cela signifie que le déchargement des calculs permet aux appareils mobiles d'obtenir une latence plus faible. En outre, l'énergie consommée dans le schéma de déchargement partiel est nettement inférieure à celle dans le cas 'No-offload'. Par exemple, le W.C.E min-max à $\eta = 0,1s$ dans la sous-figure de gauche est égal à 0,138, 0,036, 0,029, 0,014 pour le 'No-offload', puissance d'émission fixe de '$p_k = p_{\max}/2$', '$p_k = p_{\max}$' et 'Optimal p_k' respectivement. Cela signifie que le déchargement

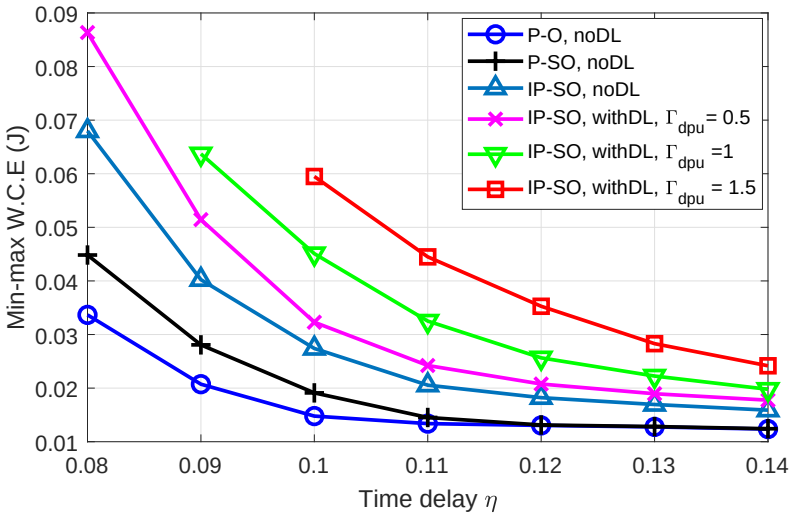

Figure 2.2 – Min-max W.C.E versus maximum allowable latency.

partiel permet d'économiser environ 5 fois l'énergie sans optimisation de la puissance de transmission et d'économiser environ 10 fois l'énergie avec une puissance de transmission optimale. De plus, la différence d'énergie consommée entre les schémas de déchargement et de non-déchargement augmente pour un plus petit nombre de bits de transmission.

La figure 2.2 présente les performances obtenues des différents scénarios de conception considérés dans cet article: solution optimale avec P-CSI - pas de données de liaison descendante ('PO, noDL'), solution avec P-CSI - pas de données de liaison descendante ('P-SO, noDL ') et IP-CSI - pas de liaison descendante données ('IP-SO, noDL'). Nous considérons également différents scénarios d'application avec une petite, moyenne et grande quantité de données de liaison descendante en comparaison avec une quantité de données de liaison montante où les performances de notre algorithme de faible complexité pour le scénario IP-CSI sont étudiées. Plus précisément, nous définissons le rapport entre la quantité de données de liaison descendante et la quantité de données de liaison montante (Γ_{dpu}) (c.-à-d. calculé comme $\sum_{l_k \in \mathcal{L}_k} b^{\mathsf{dl}}_{k,l_k} / \sum_{l_k \in \mathcal{L}_k} b_{k,l_k}$) égal à 0,5, 1 et 1,5 correspondant aux données de liaison descendante faible ('$\Gamma_{\mathsf{dpu}} = 0.5$'), aux données de liaison descendante moyenne ('$\Gamma_{\mathsf{dpu}} = 1$'), et de grandes données de liaison descendante ('$\Gamma_{\mathsf{dpu}} = 1.5$'), respectivement.

On peut observer à partir de cette figure que l'algorithme de faible complexité atteint des performances presque optimales lorsque la contrainte de temps délai maximale est moins stricte (courbes 'bleues' et 'noires'). Pour le scénario IP-CSI, les utilisateurs mobiles auront besoin de plus d'énergie pour la transmission de données afin de compenser les erreurs d'estimation CSI. Lorsque la quantité de données de liaison descendante augmente, il faut plus de temps pour transférer les

données de téléchargement, ce qui signifie que moins de temps est disponible pour télécharger les données de liaison montante et le calcul sur le serveur en le nuage. Dans certains cas, l'augmentation de la puissance de transmission à sa valeur maximale peut ne pas conduire à une amélioration du SINR, et le faible taux de transmission peut empêcher la transmission de données de liaison montante réussie dans le processus de déchargement. Dans tous les scénarios étudiés, même pour la valeur élevée de Γ_{dpu}, le schéma de déchargement partiel nous permet d'économiser considérablement de l'énergie.

2.2.2 Compression des Données et Déchargement des Calculs Conjointe dans les Systèmes Informatiques de brouillard-nuages Hiérarchiques

Au meilleur de nos connaissances, la conception de la compression des données, du déchargement de calcul, et de l'allocation des ressources conjointe pour les systèmes informatiques de brouillard-nuages hiérarchiques n'a pas été considérée dans la littérature existante. Les principales contributions de ce travail de recherche peuvent être résumées comme suit:

- Nous proposons un modèle de calcul non linéaire qui peut être ajusté pour capturer avec précision la charge de calcul encourue par la compression des données et la décompression.
- Pour la compression des données uniquement pour les utilisateurs mobiles, nous formulons la conception conjointe équitable du ratio de compression, du déchargement des calculs, et de l'allocation des ressources en tant que problème d'optimisation MINLP pour minimiser le coût pondéré maximum en énergie et en délai de service (WEDC) de tous les utilisateurs. Nous proposons un algorithme optimal, dénommé compression des données, déchargement des calculs et allocation des ressources conjointe (JCORA), qui résout ce problème difficile de manière optimale.
- Nous étudions ensuite une conception plus générale où la compression des données est effectué à la fois chez les utilisateurs mobiles et le serveur de brouillard (avec avec des ratios de compression différents). Nous proposons trois algorithmes de solution différents, à savoir l'algorithme PLA (l'approximation basé sur linéaire par morceaux), l'algorithme en deux étapes basé sur λ-recherche unidimensionnel (OSTS) et l'itératif λ-update basé sur un algorithme en deux étapes (IUTS), pour résoudre ce problème plus général.

2.2.2.1 Modèle de Système

Nous considérons un système informatique de brouillard-nuage hiérarchique composé de K utilisateurs mobiles à une seule antenne, d'un serveur de nuage et d'un serveur de brouillard colocalisés avec une station de base (BS) équipée d'un grand nombre d'antennes. Dans ce système, la BS communique avec les utilisateurs via des liaisons sans fil tandis qu'une liaison d'amenée (filaire) est déployée entre la BS colocalisée avec le serveur de brouillard et le serveur de nuage. Pour plus de commodité, nous désignons l'ensemble d'utilisateurs par $\mathcal{K}$. Nous supposons que chaque utilisateur k doit exécuter une application nécessitant c_k cycles CPU dans un intervalle de T_k^{max} secondes, où $c_{k,0}$ cycles CPU doit être exécuté localement sur l'appareil mobile et les cycles CPU restants déchargeables $c_{k,1}$ peuvent être traités localement ou déchargés et traités sur le serveur de brouillard-nuage pour économiser de l'énergie et améliorer les délais. Soit b_k^{in} le nombre de bits représentant les données encourues correspondantes des cycles CPU $c_{k,1}$ éventuellement déchargés. Une fois que $c_{k,1}$ cycles CPU sont déchargés, l'utilisateur k compresse d'abord les bits b b_k^{in} correspondants vers le bas à $b_k^{\mathsf{out,u}}$ bits avant de les envoyer au serveur de brouillard distant. Le ratio de compression est noté $\omega_k^{\mathsf{u}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,u}}$.

a) Modèle de Compression des Données:

Nous adoptons une approche pratique d'ajustement des données pour modéliser la charge de calcul de compression, la charge de calcul de décompression, et la qualité de la compression en tant que fonctions non linéaires du ratio de compression, comme suit:

$$c_k^{\mathsf{x,u}} = \gamma_{k,0}^{\mathsf{u}}\left[\gamma_{k,1}^{\mathsf{x,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{x,u}}} + \gamma_{k,3}^{\mathsf{x,u}}\right], \text{ for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{2.20}$$

$$q_k^{\mathsf{qu,u}} = \gamma_{k,3}^{\mathsf{qu,u}} - \left[\gamma_{k,1}^{\mathsf{qu,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{qu,u}}}\right], \text{ for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{2.21}$$

où 'x' = 'co' et 'de' 'représentent respectivement la compression et la décompression, $[\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}]$ représente la plage possible de ω_k^{u} et dépend de la l'algorithme de compression utilisé par l'utilisateur k, $c_k^{\mathsf{co,u}}$ et $c_k^{\mathsf{de,u}}$ indique les cycles CPU supplémentaires à la source et à la destination nécessaires pour la compression et la décompression , respectivement; $q_k^{\mathsf{qu,u}}$ représente la QoS perçue (c'est-à-dire que ce paramètre, qui n'est pris en compte que pour la compression avec perte, mesure l'écart

entre les vraies données et les données décompressées); $\gamma_{k,0}^{\mathsf{u}}$ est le nombre maximum de cycles CPU; $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i = 1, 2, 3$, sont des paramètres constants où $\gamma_{k,1}^{\mathsf{co/de/qu,u}}, \gamma_{k,3}^{\mathsf{co/de/qu,u}} \geq 0$.

b) Modèle de Calcul et de Déchargement: Nous introduisons maintenant les variables de décision de déchargement binaire s_k^{u}, s_k^{f}, et s_k^{c} pour la tâche de calcul de l'utilisateur k, où $s_k^{\mathsf{u}} = 1$, $s_k^{\mathsf{f}} = 1$, et $s_k^{\mathsf{c}} = 1$ désignent les scénarios où l'application est exécutée sur l'appareil mobile, le serveur de brouillard et le serveur de nuage, respectivement; et ces variables sont nulles sinon. De plus, nous supposons que les cycles CPU $c_{k,1}$ doivent être exécutés à exactement un emplacement, ce qui implique $s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 1$. Ensuite, la charge de calcul totale de l'utilisateur k sur l'appareil mobile, notée c_k^{u}, et sur le serveur de brouillard, notée c_k^{f}, sont donnés respectivement, $c_k^{\mathsf{u}} = c_{k,0} + s_k^{\mathsf{u}} c_{k,1} + (1 - s_k^{\mathsf{u}}) c_k^{\mathsf{co,u}}$ et $c_k^{\mathsf{f}} = s_k^{\mathsf{f}} \left(c_{k,1} + c_k^{\mathsf{de,u}} \right)$.

L'énergie de calcul locale consommée par l'utilisateur k et le temps de calcul local peuvent être exprimés, respectivement, comme $\xi_{1,k}^{\mathsf{u}} = \alpha_k {f_k^{\mathsf{u}}}^2 c_k^{\mathsf{u}}$ et $t_{1,k}^{\mathsf{u}} = c_k^{\mathsf{u}} / f_k^{\mathsf{u}}$, où f_k^{u} est la vitesse d'horloge du processeur de l'utilisateur k et α_k indique le coefficient d'énergie spécifié par le modèle de processeur [28]. Soit f_k^{f} la vitesse d'horloge du processeur utilisé sur le serveur de brouillard pour traiter $c_{k,1}$. Ensuite, le temps de calcul sur le serveur de brouillard est donné par $t_{1,k}^{\mathsf{f}} = c_k^{\mathsf{f}} / f_k^{\mathsf{f}}$. Nous supposons que la tâche de calcul de chaque utilisateur est exécutée sur le serveur de nuage avec un délai fixe de T^{c} secondes.

c) Modèle de Communication:

Nous supposons que l'estimation du canal est parfaite et que ZF est appliqué à la BS, alors le taux de liaison montante moyen de l'utilisateur k à la BS (serveur de brouillard) est exprimé comme $r_k = \rho_k \log_2(1 + P_k \beta_{k,0})$, où P_k est la puissance d'émission en liaison montante par Hz de l'utilisateur k, ρ_k désigne la bande passante de transmission et $\beta_{k,0} = M_0 \beta_k / \sigma_{\mathsf{bs}}$. Ici, β_k représente le coefficient d'évanouissement à grande échelle, σ_{bs} est la densité de puissance de bruit (watts par Hz) et M_0 est le gain de formation de faisceau à entrées multiples et sorties multiples (MIMO) [25]. On suppose que le nombre d'antennes est suffisamment grand pour que M_0 soit identique pour tous les utilisateurs. Ensuite, le temps et l'énergie de transmission en liaison montante de l'utilisateur k peuvent être calculés, respectivement, comme $t_{2,k}^{\mathsf{u}} = (1 - s_k^{\mathsf{u}}) b_k^{\mathsf{out,u}} / r_k$ et $\xi_{2,k}^{\mathsf{u}} = \rho_k (P_k + P_{k,0}) t_{2,k}^{\mathsf{u}}$, où $P_{k,0}$ représente la consommation électrique du circuit par Hz. Pour la transmission de données entre le serveur de brouillard et le serveur de nuage, une liaison d'amenée avec une capacité D^{max} bps (bits par seconde) est supposée. Soit d_k le débit d'amenée alloué à l'utilisateur k. Ensuite,

le temps de transmission du serveur de brouillard vers le serveur de nuage est $t^{\mathsf{f}}_{2,k} = s^{\mathsf{c}}_k b^{\mathsf{out,u}}_k / d_k$. Ensuite, le délai total pour terminer la tâche de calcul de l'utilisateur k est donné par

$$T_k = t^{\mathsf{u}}_{1,k} + t^{\mathsf{u}}_{2,k} + t^{\mathsf{f}}_{1,k} + t^{\mathsf{f}}_{2,k} + s^{\mathsf{c}}_k T^{\mathsf{c}}. \tag{2.22}$$

De plus, l'énergie globale consommée par l'utilisateur k pour le traitement de sa tâche est donnée par

$$\xi_k = \xi^{\mathsf{u}}_{1,k} + \xi^{\mathsf{u}}_{2,k}. \tag{2.23}$$

Pratiquement, tous les utilisateurs souhaitent économiser de l'énergie et profiter d'une faible latence d'exécution des applications. Par conséquent, nous adoptons le WEDC comme fonction objectif de chaque utilisateur k comme $\Xi_k = w^{\mathsf{T}}_k T_k + w^{\mathsf{E}}_k \xi_k$, où w^{T}_k et w^{E}_k représentent les poids correspondant respectivement à la latence du service et à l'énergie consommée. Ces poids peuvent être prédéterminés par les utilisateurs pour refléter leurs priorités ou leurs intérêts. La conception proposée vise à minimiser la fonction WEDC pour chaque utilisateur en maintenant l'équité entre tous les utilisateurs. À cette fin, nous considérons le problème d'optimisation min-max suivant:

$$
\begin{aligned}
(\mathcal{P}_2) \quad \min_{\Omega_1 \cup \eta} \quad & \eta && \text{(2.24a)}\\
\text{s. t.} \quad & \Xi_k \le \eta, \forall k, && \text{(2.24b)}\\
& f^{\mathsf{u}}_k \le F^{\mathsf{max}}_k, \forall k, && \text{(2.24c)}\\
& \sum_k f^{\mathsf{f}}_k \le F^{\mathsf{f,max}}, && \text{(2.24d)}\\
& s^{\mathsf{u}}_k, s^{\mathsf{f}}_k, s^{\mathsf{c}}_k \in \{0,1\}, \forall k, && \text{(2.24e)}\\
& s^{\mathsf{u}}_k + s^{\mathsf{f}}_k + s^{\mathsf{c}}_k = 1, \forall k, && \text{(2.24f)}\\
& \omega^{\mathsf{u,min}}_k \le \omega^{\mathsf{u}}_k \le \omega^{\mathsf{u,max}}_k, \forall k, && \text{(2.24g)}\\
& 0 \le \rho_k P_k \le p^{\mathsf{max}}_k, \forall k, && \text{(2.24h)}\\
& 0 \le \rho_k \le \rho^{\mathsf{max}}_k, \ \forall k, && \text{(2.24i)}\\
& \textstyle\sum_k d_k \le D^{\mathsf{max}}, , && \text{(2.24j)}\\
& T_k \le T^{\mathsf{max}}_k, \forall k, && \text{(2.24k)}
\end{aligned}
$$

Algorithm 2.4. Optimal Joint DC, Offloading, and Resource Allocation (JCORA)

1: **Initialize**: Compute $\eta_k^{\mathsf{lo}}, \forall k \in \mathcal{K}$ as in (2.25), choose ϵ, assign $\eta^{\mathsf{min}} = 0$, $\eta^{\mathsf{max}} = \max_k(\eta_k^{\mathsf{lo}})$, and set BOOL = *False*.
2: **while** $(\eta^{\mathsf{max}} - \eta^{\mathsf{min}} > \epsilon)$ & (BOOL = *False*) **do**
3: Assign $\eta = (\eta_{\mathsf{max}} + \eta_{\mathsf{min}})/2$, and then define sets $\mathcal{A} = \{k|\eta_k^{\mathsf{lo}} \leq \eta\}$ and $\mathcal{B} = \mathcal{K}/\mathcal{A}$.
4: Check feasibility of $(\mathcal{P}_\mathcal{B})$ as in Section 2.2.2.2.
5: **if** $(\mathcal{P}_\mathcal{B})$ *is feasible* **then** $\eta^{\mathsf{max}} = \eta$, BOOL = *True*, **else** $\eta^{\mathsf{min}} = \eta$, BOOL = *False*, **end if**
6: **end while**

où $\Omega_1 = \cup_{k\in\mathcal{K}}\Omega_{1,k}$, $\Omega_{1,k} = \{s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, \omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, P_k, \rho_k, d_k\}$; F_k^{max} est la vitesse d'horloge maximale du processeur de l'utilisateur k, $F^{\mathsf{f,max}}$ est la vitesse maximale du processeur du serveur de brouillard, p_k^{max} est la puissance d'émission maximale de l'utilisateur k, $[\omega_k^{\mathsf{u,min}}, \omega_k^{\mathsf{u,max}}]$ indique la plage possible du ratio de compression ω_k^{u} qui peut garantir la qualité de service requise des données récupérées, ρ_k^{max} est le coût de service contraint maximum. En particulier, pour la compression de données sans perte où la QoS perçue $q_k^{\mathsf{qu,u}} = 1$ pour tous ω_k^{u}, cette plage réalisable est déterminée comme $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ et $\omega_k^{\mathsf{u,max}} = \omega_{k,1}^{\mathsf{u,max}}$. Pour la compression de données avec perte où la QoS perçue doit être supérieure à $q_k^{\mathsf{qu,u,min}}$, cette plage est déterminée comme $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ et $\omega_k^{\mathsf{u,max}} = \min\left\{\omega_{k,1}^{\mathsf{u,max}}, \left((\gamma_{k,3}^{\mathsf{qu,u}} - q_k^{\mathsf{qu,u,min}})/\gamma_{k,1}^{\mathsf{qu,u}}\right)^{1/\gamma_{k,2}^{\mathsf{qu,u}}}\right\}$.

Soit η_k^{lo} le WEDC minimum de UE k lorsqu'il exécute sa tâche de calcul localement, qui est calculée à partir de $(\mathcal{P}_2)$ comme

$$\eta_k^{\mathsf{lo}} = \begin{cases} \mathcal{Q}_{k,0}(f_k^{\mathsf{u,sta}}), \text{ if } f_k^{\mathsf{u,sta}} \in [f_k^{\mathsf{u,min}}, F_k^{\mathsf{max}}] \\ \min\left(\mathcal{Q}_{k,0}(f_k^{\mathsf{u,min}}), \mathcal{Q}_{k,0}(F_k^{\mathsf{max}})\right), \text{ otherwise,} \end{cases} \tag{2.25}$$

où $\mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) = w_k^{\mathrm{E}}\alpha_k(f_k^{\mathsf{u}})^2 c_k + w_k^{\mathrm{T}} c_k / f_k^{\mathsf{u}}$, $f_k^{\mathsf{u,min}} = c_k/T_k^{\mathsf{max}}$, et $f_k^{\mathsf{u,sta}} = \sqrt[3]{w_k^{\mathsf{T}}/(2w_k^{\mathsf{E}}\alpha_k)}$.

On peut vérifier que si $\eta^\star$ est la valeur objective optimale du problème $(\mathcal{P}_2)$, alors une classification optimale, $(\mathcal{A}^\star, \mathcal{B}^\star)$, peut être déterminée comme $\mathcal{A}^\star = \{k|\eta_k^{\mathsf{lo}} \leq \eta^\star\}$, and $\mathcal{B}^\star = \mathcal{K}\backslash\mathcal{A}^\star$, où $\mathcal{A}$ et $\mathcal{B}$ sont respectivement les ensembles d'utilisateurs exécutant localement et déchargeant. Par conséquent, nous proposons l'algorithme 2.4, nommé JCORA, pour s'attaquer à $(\mathcal{P}_2)$.

Dans cet algorithme, nous calculons initialement η_k^{lo} pour tous les utilisateurs dans $\mathcal{K}$ comme dans (2.25). Ensuite, nous utilisons la recherche de bissection pour trouver le $\eta^\star$ optimal où la borne supérieure η^{max} et la borne inférieure η^{min} sont mises à jour de manière itérative jusqu'à ce que la différence entre eux devienne suffisamment petite, $(\mathcal{P}_\mathcal{B})$ est faisable et les ensembles $\mathcal{A}$ et $\mathcal{B}$ ne

changent pas. Il est à noter que $(\mathcal{P}_{\mathcal{B}})$ est $(\mathcal{P}_2)$ pour les utilisateurs de l'ensemble $\mathcal{B}$. La vérification de faisabilité de $(\mathcal{P}_{\mathcal{B}})$ est présentée comme suit.

2.2.2.2 Vérification de Faisabilité de $(\mathcal{P}_{\mathcal{B}})$

Afin de vérifier la faisabilité de $(\mathcal{P}_{\mathcal{B}})$, nous considérons le problème suivant

$$(\mathcal{P}_{\mathsf{FV},\eta}) \quad \min_{\Omega_{\mathcal{B}}} \sum_{k\in\mathcal{B}} f_k^{\mathsf{f}} \quad \text{s. t.} \quad (2.24b), (2.24c), (2.24e) - (2.24k).$$

Ce problème minimise la ressource de calcul totale requise du serveur de brouillard soumise à toutes les contraintes de $(\mathcal{P}_{\mathcal{B}})$ sauf $(2.24d)$. Soit $G^{\star}_{\mathcal{B},\eta}$ la valeur objectif du problème $(\mathcal{P}_{\mathsf{FV},\eta})$. Ensuite, la faisabilité de $(\mathcal{P}_{\mathcal{B}})$ peut être vérifiée en comparant $G^{\star}_{\mathcal{B},\eta}$ avec $F^{\mathsf{f},\mathsf{max}}$. En particulier, le problème $(\mathcal{P}_{\mathcal{B}})$ est réalisable si $G^{\star}_{\mathcal{B},\eta} \leq F^{\mathsf{f},\mathsf{max}}$. Sinon, $(\mathcal{P}_{\mathcal{B}})$ est irréalisable. Pour résoudre $(\mathcal{P}_{\mathsf{FV},\eta})$, nous considérons deux sous-problèmes suivants

$$\begin{aligned}
&(\mathcal{P}_3)_k \quad \min_{\Omega_{2,k}} f_k^{\mathsf{f}} \quad \text{s. t.} \quad s_k^{\mathsf{f}} = 1,\ (2.24b)_k,\ (2.24c)_k,\ (2.24g)_k - (2.24i)_k,\ (2.24k)_k,\\
&(\mathcal{P}_4)_k \quad \min_{\Omega_{2,k}\cup d_k\setminus f_k^{\mathsf{f}}} d_k \quad \text{s. t.} \quad s_k^{\mathsf{c}} = 1, (2.24b)_k,\ (2.24c)_k,\ (2.24g)_k - (2.24i)_k,\ (2.24k)_k,
\end{aligned}$$

où $\Omega_{2,k} = \{\omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, p_k, \rho_k\}$, $(2.24b)_k$, $(2.24c)_k$, $(2.24g)_k - (2.24i)_k$, et $(2.24k)_k$ dénotent les contraintes respectives de l'utilisateur k correspondant à $(2.24b)$, $(2.24c)$, $(2.24g) - (2.24i)$, et $(2.24k)$.

Dans les sous-problèmes $(\mathcal{P}_3)_k$ et $(\mathcal{P}_4)_k$, posons $\tilde{\omega}_k^{\mathsf{u}} = \log(\omega_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{u}} = \log(f_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{f}} = \log(f_k^{\mathsf{f}})$, $\tilde{P}_k = \log(P_k)$, et $\tilde{\rho}_k = \log(\rho_k)$, nous prouvons que $(\mathcal{P}_3)_k$ est convexe par rapport à l'ensemble $\tilde{\Omega}_{2,k}\cup\tilde{l}_k$, où $\tilde{l}_k = \tilde{\omega}_k^{\mathsf{u}} + \tilde{\rho}_k$ et $\tilde{\Omega}_{2,k} = \{\tilde{\omega}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{f}}, \tilde{P}_k, \tilde{\rho}_k\}$. De même, $(\mathcal{P}_4)_k$ peut être converti en un problème convexe via une transformation logarithmique. Si $(\mathcal{P}_3)_k$ est impossible, nous définissons $s_k^{\mathsf{c}} = f$, et si $(\mathcal{P}_4)_k$ est impossible, nous définissons $s_k^{\mathsf{c}} = 0$. Avec l'objectif optimal obtenu de $(\mathcal{P}_3)_k$ et $(\mathcal{P}_4)_k$, noté $f_k^{\mathsf{f},\mathsf{rq}}$ et d_k^{rq}, respectivement, le problème qui est l'équivalent du problème $(\mathcal{P}_{\mathsf{FV},\eta})$, est défini comme

Algorithm 2.5. Feasibility Verification of $(\mathcal{P}_{\mathcal{B}})$

1: Solve $(\mathcal{P}_3)_k$ to find $f_k^{\mathsf{f},\mathsf{rq}}, \forall k \in \mathcal{B}$.
2: Solve $(\mathcal{P}_4)_k$ to find $d_k^{\mathsf{rq}}, \forall k \in \mathcal{B}$.
3: **if** $\exists k$ such that $s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 0$ **then**
4: Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible
5: **else**
6: Solve $(\mathcal{P}_{\mathsf{FV},\eta})$ to find $G^{\star}_{\mathcal{B},\eta}$.
7: **if** $G^{\star}_{\mathcal{B},\eta} < F^{\mathsf{f},\mathsf{max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible **end if**
8: **end if**

$$(\mathcal{P}_{\mathsf{FV},\eta}) \quad \min_{\Omega_3} \ \mathcal{G}_{\mathcal{B},\eta}(\Omega_3) = \sum_{k\in\mathcal{B}} (1 - s_k^{\mathsf{c}}) f_k^{\mathsf{f},\mathsf{rq}}$$
$$\text{s. t.} \ \sum_{k\in\mathcal{B}} s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \leq D^{\mathsf{max}}, \ s_k^{\mathsf{c}} \in \{0,1\},$$

où $\Omega_3 = \{s_k^{\mathsf{c}} | k \in \mathcal{B}\}$ pour un η donné. En fait, $(\mathcal{P}_{\mathsf{FV},\eta})$ est un problème *"0-1 knapsack"* [29], qui peut être résolu de manière optimale et en utilisant efficacement le solveur CVX. Si $G^{\star}_{\mathcal{B},\eta} \leq F^{\mathsf{f},\mathsf{max}}$, en combinant l'ensemble de toutes les solutions de $(\mathcal{P}_3)_k$'s, $(\mathcal{P}_4)_k$'s, et $(\mathcal{P}_{\mathsf{FV},\eta})$ donne une solution réalisable de $(\mathcal{P}_{\mathcal{B}})$ pour cette valeur de η. Par conséquent, $(\mathcal{P}_{\mathcal{B}})$ est réalisable dans un tel scénario. La vérification de faisabilité de $(\mathcal{P}_{\mathcal{B}})$ est résumée dans l'algorithme 2.5.

Theorem 2.1. *L'intégration de l'algorithme 2.5 dans l'algorithme 2.4 donne l'optimum global de MINLP* $(\mathcal{P}_2)$.

Proof. L'algorithme 2.5 vérifie la faisabilité de $(\mathcal{P}_{\mathcal{B}})$ pour une valeur donnée de $\eta_{\mathcal{B}} = \eta$. Par conséquent, si l'algorithme 2.4 utilise l'algorithme 2.5, $(\mathcal{P}_2)$ est résolu de manière optimale. Notez qu'après convergence, les variables optimales sont données par la solution optimale de $(\mathcal{P}_3)_k$ si $s_k^{\mathsf{f}} = 1$ ou $(\mathcal{P}_4)_k$ si $s_k^{\mathsf{c}} = 1$ où les valeurs des s_k^{f} et s_k^{c} sont les résultats de $(\mathcal{P}_{\mathsf{FV},\eta})$.

□

2.2.2.3 Compression de Données chez les Utilisateurs Mobiles et le Serveur Fog

Nous considérons maintenant le cas plus général où le serveur de brouillard a également effectué la compression de données avant de transmettre les données compressées via la liaison d'amenée au serveur cloud. Cette option de conception peut encore améliorer les performances des systèmes

avec une liaison d'amenée congestionnée. Le ratio de compression d'amenée est défini comme $\omega_k^{\mathsf{f}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,f}}$ où $b_k^{\mathsf{out,f}}$ représente le nombre de bits transmis sur le lien d'amenée. Notons s_k^{m} comme variable binaire indiquant si la compression de données est executée ou non sur le serveur de brouillard pour l'utilisateur k ($s_k^{\mathsf{m}} = 1$ pour la compression de données et $s_k^{\mathsf{m}} = 0$, sinon). Dans ce cas général, les contraintes(2.24e) et (2.24f) peuvent être réécrites sous la forme

$$s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}} \in \{0,1\}, \forall k \in \mathcal{K}, \tag{2.26a}$$

$$s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + s_k^{\mathsf{m}} = 1, \forall k \in \mathcal{K}, \tag{2.26b}$$

Ensuite, la charge de calcul pour la compression et les données de sortie correspondant au *Mode 3* peuvent être modélisées comme $c_k^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}}\left[\gamma_{k,1}^{\mathsf{co,f}}(\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \gamma_{k,3}^{\mathsf{co,f}}\right]$ et $b_k^{\mathsf{out,f}} = b_k^{\mathsf{in}}/\omega_k^{\mathsf{f}}$, respectivement, où $\gamma_{k,0}^{\mathsf{f}}, \gamma_{k,1}^{\mathsf{co,f}}, \gamma_{k,3}^{\mathsf{co,f}} \in \mathbb{R}_+$ sont des nombres positifs. Ici, nous avons des contraintes supplémentaires pour le ratio de compression sur le serveur de brouillard comme

$$\omega_k^{\mathsf{f}} \in [\omega_k^{\mathsf{f,min}}, \omega_k^{\mathsf{f,max}}],\ \forall k \in \mathcal{K}. \tag{2.27}$$

Ensuite, la charge de calcul totale pour l'utilisateur k sur le serveur de brouillard devient $\check{c}_k^{\mathsf{f}} = s_k^{\mathsf{f}}\left(c_{k,1} + c_k^{\mathsf{de,u}}\right) + s_k^{\mathsf{m}}(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})$, et le temps de calcul sur le serveur de brouillard est $\check{t}_{1,k}^{\mathsf{f}} = \check{c}_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. De plus, le temps de transmission engendré par le déchargement des données de l'utilisateur k du serveur de brouillard vers le serveur de nuage peut être réécrit comme $\check{t}_{2,k}^{\mathsf{f}} = \left(s_k^{\mathsf{f}} b_k^{\mathsf{out,u}} + s_k^{\mathsf{m}} b_k^{\mathsf{out,f}}\right)/d_k$. Ensuite, le délai total pour terminer la tâche de calcul de l'utilisateur k est donné par $\check{T}_k = t_{1,k}^{\mathsf{u}} + t_{2,k}^{\mathsf{u}} + \check{t}_{1,k}^{\mathsf{f}} + \check{t}_{2,k}^{\mathsf{f}} + (s_k^{\mathsf{c}} + s_k^{\mathsf{m}})T^{\mathsf{c}}$, et le WEDC devient $\check{\Xi}_k = w_k^{\mathsf{T}}\check{T}_k + w_k^{\mathsf{E}}\xi_k$. Ensuite, la contrainte (2.24k) est réécrite comme

$$\check{T}_k \leq T_k^{\mathsf{max}}. \tag{2.28}$$

Les versions étendues du problème $(\mathcal{P}_2)$ peuvent être déclarées comme

$$(\mathcal{P}_2^{\mathsf{ext}}) \quad \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\} \cup \eta} \quad \eta$$

$$\text{s. t.} \quad \check{\Xi}_k \leq \eta, \tag{2.29a}$$

$$(2.24c), (2.24d), (2.24g)-(2.24j), (2.26a), (2.26b), (2.27), (2.28).$$

Algorithm 2.6. PLA-based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **Initialize**: L, η
2: Compute $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all $k \in \mathcal{B}$ as in Step 1 and 2 of Algorithm 2.5.
3: Define $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L,\ \forall k \in \mathcal{B}, l = 0 : L$.
4: Compute $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$. **If** $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ is unbounded **then** Remove point $d_{k,l}$ **end if**.
5: Compute $A_{k,l}$, $B_{k,l}$, and then solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$ to get optimal value $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star}$ of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$.
6: **if** $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star} \leq F^{\mathsf{f,max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible **end if**

Pour résoudre le problème étendu, nous utilisons l'approche de solution générale présentée dans la section précédente. Maintenant, nous présentons les méthodes de vérification de faisabilité pour $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$.

a) L'Algorithme basé sur l'Approximation Linéaire par Morceaux (PLA):

Après avoir déterminé $f_k^{\mathsf{f,rq}}$ et d_k^{rq}, respectivement, nous déterminons les ressources de calcul de brouillard requises pour un $d_k \in (0, d_k^{\mathsf{rq}})$ donnée en résolvant le problème suivant:

$$(\mathcal{P}_{d_k}) \quad \min_{\Omega_{2,k} \cup \{\omega_k^{\mathsf{f}}\}} f_k^{\mathsf{f}}$$
$$\text{s. t. } s_k^{\mathsf{m}} = 1,\ (2.29a)_k,\ (2.24c)_k,\ (2.24g)_k - (2.24i)_k,\ (2.28)_k,\ (2.27)_k.$$

Soit $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ la solution optimale de ce problème, qui peut être obtenue en utilisant les transformations logarithmiques décrites dans la section 2.2.2.2. Cependant, trouver une expression de forme fermée pour $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ est intraitable. Par conséquent, nous proposons d'utiliser la méthode PLA pour diviser le domaine d'origine en plusieurs petits segments tels que $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ peut être approximée par une fonction linéaire dans chaque segment.

Supposons que l'intervalle $[\epsilon_{\mathsf{d}}, d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}}]$ est divisé en segments L de taille égale, où ϵ_{d} est un très petit nombre par rapport à d_k^{rq}, par exemple, $\epsilon_{\mathsf{d}} = 1$. Plus précisément, le segment l correspond à l'intervalle $[d_{k,l}, d_{k,l+1}]$, où $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L$ est un point tel que $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ et la valeur de la fonction approchée à ce point sont égales. Ensuite, nous pouvons approximer $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ comme $\hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}\Big(V_k, U_k\Big) = \sum_{l=0}^{L-1} \left(v_{k,l} A_{k,l} + u_{k,l} B_{k,l}\right)$, où $A_{k,l} = \Big(\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l+1}) - \mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})\Big)/(d_{k,l+1} - d_{k,l})$, $B_{k,l} = \mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l}) - A_{k,l} d_{k,l}$, $V_k = \{v_{k,l}, l = 0, 1, ..., L-1\}$, $U_k = \{u_{k,l}, l = 0, 1, ..., L-1\}$, et la variable continue $v_{k,l}$ et la variable binaire $u_{k,l}$ satisfont aux contraintes suivantes:

$$s_k^{\mathsf{m}} = \sum_{l=0}^{L-1} u_{k,l} \leq 1, \forall k \in \mathcal{B}, \tag{2.30}$$

$$u_{k,l} d_{k,l} \leq v_{k,l} \leq u_{k,l+1} d_{k,l+1}, \forall k \in \mathcal{B}, l = 0, 1, ..., L-1. \tag{2.31}$$

Ensuite, nous avons $s_k^{\mathsf{m}} d_k = \sum_{l=0}^{L-1} v_{k,l}$. Par conséquent, le problème $(\mathcal{P}_{\mathsf{FV},\eta})$, qui est utilisé pour déterminer le total minimal requis des ressources de calcul du brouillard pour tous les utilisateurs, est modifié dans ce cas étendu comme suit:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}}) \quad \min_{\check{\Omega}_3} \quad \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{PLA}}\left(\check{\Omega}_3\right) = \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} + \hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}\left(V_k, U_k\right)\right)$$

$$\text{s. t.} \quad s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, u_{k,l} \in \{0,1\}, \ \forall k, l, \tag{2.32a}$$

$$s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + \sum\nolimits_{l=0}^{L-1} u_{k,l} = 1, \tag{2.32b}$$

$$u_{k,l} d_{k,l} \leq v_{k,l} \leq u_{k,l+1} d_{k,l+1}, \ \forall k, l, \tag{2.32c}$$

$$\sum\nolimits_{k\in\mathcal{B}} \left(\sum\nolimits_{l=0}^{L-1} v_{k,l} + s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \right) \leq D^{\mathsf{max}}, \tag{2.32d}$$

où $\check{\Omega}_3 = \cup_{k\in\mathcal{B}} \left(s_k^{\mathsf{f}} \cup s_k^{\mathsf{c}} \cup U_k \cup V_k \right)$ et contraintes (2.32a), (2.32b), et (2.32c)-(2.32d) sont les contraintes transformées des contraintes d'origine $(2.26a)$, $(2.26b)$, et $(2.24j)$, respectivement. Ce problème transformé est un problème MILP, qui peut être résolu efficacement en utilisant le solveur CVX. L'algorithme basé sur le PLA pour vérifier la faisabilité de $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ est résumé dans l'algorithme 2.6, qui peut être intégré dans l'algorithme 2.4 pour résoudre $(\mathcal{P}_2^{\mathsf{ext}})$. Il est à noter que si la valeur de $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ est illimitée pour un $d_{k,l}$, ce point irréalisable est supprimé lors de l'application de l'algorithme basé sur PLA.

b) Approche de Solution en Deux Etapes (TSA)

Dans cette section, deux algorithmes en deux étapes sont développés en exploitant le fait que la charge de calcul de décompression (et donc la consommation d'énergie associée) est presque indépendante du ratio de compression. Cela implique que pour un η donné, les valeurs optimales f_k^{u}, ω_k^{u}, p_k, et ρ_k pour l'utilisateur mobile k sont similaire pour $s_k^{\mathsf{f}} = 1$ et $s_k^{\mathsf{c}} = 1$. Par conséquent, dans la première étape, après avoir résolu $(\mathcal{P}_3)_k$ et $(\mathcal{P}_4)_k$, $\forall k \in \mathcal{B}$, nous pouvons définir ces variables à la so-

lution optimale correspondante de $(\mathcal{P}_3)_k$, notée $f_{k,1}^{\mathsf{u}\star}$, $\omega_{k,1}^{\mathsf{u}\star}$, $p_{k,1}^{\star}$, et $\rho_{k,1}^{\star}$. Dans la deuxième étape, nous trouvons les variables restantes relatives au serveur de brouillard $\Omega_4 = \cup_{k\in\mathcal{B}}\{s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}}, d_k, f_k^{\mathsf{f}}, \omega_k^{\mathsf{f}}\}$ en résolvant le problème suivant [1]

$$
\begin{aligned}
(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}}) \quad \min_{\Omega_4} \quad & \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{TSA}}(\Omega_4) = \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{m}} f_k^{\mathsf{f}} + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \right) \\
\text{s. t.} \quad & s_k^{\mathsf{m}} \left(\frac{b_k^{\mathsf{out,f}}}{d_k} + \frac{(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})}{f_k^{\mathsf{f}}} \right) \leq \nu_{k,0}, && \text{(2.33a)} \\
& \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{m}} d_k + s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \right) \leq D^{\mathsf{max}}, && \text{(2.33b)} \\
& (2.26a), (2.26b), (2.27),
\end{aligned}
$$

où $\nu_{k,0} = \min\{(\eta - \Xi_{k,1})/w_k^{\mathsf{T}}, T_k^{\mathsf{max}} - T_{k,1}\} + (c_{k,1} + c_k^{\mathsf{de}})/f_k^{\mathsf{f,rq}} - T^{\mathsf{c}}$, et $\Xi_{k,1}$ et $T_{k,1}$ sont les valeurs optimales de Ξ_k et T_k dans $(\mathcal{P}_3)_k$, respectivement. Parce que $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ est un problème MINLP difficile, nous nous y attaquons en réduisant l'ensemble de variables en fonction des observations suivantes.

Observation 1: Pour toute valeur de d_k satisfaisante (2.33b), la solution optimale de f_k^{f} dans $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ peut être déterminé comme $f_k^{\mathsf{f}\star} = s_k^{\mathsf{m}} \frac{(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})}{\nu_{k,0} - b_k^{\mathsf{out,f}}/d_k} = s_k^{\mathsf{m}} \mathcal{H}_0\Big(\omega_k^{\mathsf{f}}, d_k\Big)$, où $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}}, d_k\Big) = \frac{\omega_k^{\mathsf{f}} d_k \left[\tilde{\gamma}_{k,1}^{\mathsf{co,f}} (\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}} \right]}{\nu_{k,0}\omega_k^{\mathsf{f}} d_k - b_k^{\mathsf{in}}}$, $\tilde{\gamma}_{k,1}^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}} \gamma_{k,1}^{\mathsf{co,f}}$, et $\tilde{\gamma}_{k,3}^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}} \gamma_{k,3}^{\mathsf{co,f}} + c_k^{\mathsf{de,u}}$.

Observation 2: Lorsque $s_k^{\mathsf{m}} = 1$ et $d_k \geq \bar{d}_{k,1}$, la valeur optimale de ω_k^{f}, noté $\omega_k^{\mathsf{f}\star}$, est donnée comme suit:

$$
\omega_k^{\mathsf{f}\star} = \begin{cases} \omega_k^{\mathsf{max,f}}, \text{ if } \gamma_{k,2}^{\mathsf{co,f}} \leq 0 \cup \{\gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,1} < d_k \leq \bar{d}_{k,2}\}, \\ \mathrm{inv}\Big(\mathcal{H}_1\Big(d_k\Big)\Big), \quad \text{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,2} < d_k \leq \bar{d}_{k,3}, \\ \omega_k^{\mathsf{f,min}}, \quad \text{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, d_k > \bar{d}_{k,3}, \end{cases} \tag{2.34}
$$

[1] Nous notons qu'en réduisant le nombre de variables d'optimisation dans $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$, la complexité des algorithmes résultants pour la vérification de faisabilité de $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ est inférieure à celui de l'algorithme basé sur PLA.

Algorithm 2.7. One-dimensional Search Based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **initialize**: Δ_λ, $\lambda = 0$, Assign $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible.
2: Define $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all k as in Step 2 and Step 3 of Algorithm 2.5.
3: **repeat**
4: Assign $\lambda = \lambda + \Delta_\lambda$. Compute $d_{k,\lambda}$ as in (2.35) and solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ to find $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$.
5: **if** $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \leq F^{\mathsf{f,max}}$ **then**
6: Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible; **break**
7: **end if**
8: **until** $\lambda = \lambda^{\mathsf{max}}$

où $\bar{d}_{k,1} = b_k^{\mathsf{in}}/(\nu_{k,0}\omega_k^{\mathsf{f}})$, $\bar{d}_{k,2} = \mathcal{H}_1\left(\omega_k^{\mathsf{max,f}}\right)$, $\bar{d}_{k,3} = \mathcal{H}_1\left(\omega_k^{\mathsf{f,min}}\right)$, et $\text{inv}\left(\mathcal{H}_1\left(d_k\right)\right)$ est la valeur de ω_k^{f} pour lequel $\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)$ est égal à d_k, et $\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right) \triangleq \frac{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} b_k^{\mathsf{in}}(\gamma_{k,2}^{\mathsf{co,f}}+1)\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}} b_k^{\mathsf{in}}}{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} \nu_{k,0} \gamma_{k,2}^{\mathsf{co,f}} \left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}+1}}$.

Ensuite, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ est équivalent au problème suivant::

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}}) \quad \min_{\tilde{\Omega}_4} \quad \sum_{k\in\mathcal{B}} \left[s_k^{\mathsf{m}} \mathcal{H}_0\left(\omega_k^{\mathsf{f}\star}, d_k\right) + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \right]$$
$$\text{s. t. } (2.26a), (2.26b), (2.33b),$$

où $\tilde{\Omega}_4 = \cup_{k\in\mathcal{B}}\{s_k^{\mathsf{c}}, s_k^{\mathsf{f}}, s_k^{\mathsf{m}}, d_k\}$. On peut vérifier que la valeur optimale de d_k pour $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, notée $d_k^\star$, est donnée comme suit:

$$d_k^\star = \begin{cases} 0, \quad \text{if} \quad s_k^{\mathsf{f}\star} = 1, \\ d_k^{\mathsf{rq}}, \quad \text{if} \quad s_k^{\mathsf{c}\star} = 1, \\ \left\{ d_{k,\lambda} \middle| \left(\frac{\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)}{\partial d_k}\Big|_{d_k = d_{k,\lambda}} \right) + \lambda = 0 \right\}, \text{sinon}, \end{cases} \tag{2.35}$$

où λ est le multiplicateur de contrainte de Lagrange (2.33b).

Observation 3: Le gradient $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k$ est une fonction monotone croissante de d_k.

Avec **Observation 3**, nous pouvons conclure que pour un λ donné, il existe au plus un valeur de d_k satisfaisant $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k + \lambda = 0$. Cela signifie que si le λ optimal est connu, le problème $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ peut être résolu efficacement. Par conséquent, comme décrit ci-dessous, pour résoudre $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, nous proposons deux algorithmes: l'un est basé sur la λ-recherche unidimensionnelle, et l'autre est basée sur mise à jour itérative de λ.

b1) L'Algorithme en deux étapes basé sur la λ-recherche une-dimensionnel (OSTS Alg.) Pour un λ donné, supposons que $d_{k,\lambda}$ satisfait $\partial\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k\Big|_{d_k=d_{k,\lambda}} + \lambda = 0$. En définissant $f_{k,\lambda} = \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k=d_{k,\lambda}}$, $\mu_{k,\lambda} = s_k^{\mathsf{m}}$, $\mu_{k,\lambda} = 1 - s_k^{\mathsf{c}}$, et $\mu_{k,\lambda} = s_k^{\mathsf{c}}(1-x_k)$, nous pouvons trouver la solution optimale de $\cup_{k\in\mathcal{B}}\{s_k^{\mathsf{c}}, x_k, d_k\}$ en résolvant le problème suivant:

$$
\begin{aligned}
(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda \quad & \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) = \min_{\cup_{k\in\mathcal{B}} s_{k,\lambda}} \sum_{k\in\mathcal{B}} \Big[s_{k,\lambda}^{\mathsf{m}} f_{k,\lambda} + s_{k,\lambda}^{\mathsf{f}} f_k^{\mathsf{f,rq}} \Big] \\
\text{s. t.} \quad & \sum_{k\in\mathcal{B}} s_{k,\lambda}^{\mathsf{m}} d_{k,\lambda} + (1 - s_{k,\lambda}^{\mathsf{f}} - s_{k,\lambda}^{\mathsf{m}}) d_k^{\mathsf{rq}} \le D^{\mathsf{max}}, \\
& s_{k,\lambda}^{\mathsf{m}}, s_{k,\lambda}^{\mathsf{f}} \in \{0,1\},
\end{aligned}
$$

où $s_{k,\lambda} = \{s_{k,\lambda}^{\mathsf{f}}, s_{k,\lambda}^{\mathsf{m}}\}$. Le problème transformé ci-dessus est un problème de programme linéaire entier (ILP), qui peut être résolu efficacement par CVX. Soit $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$ la valeur optimale de $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$, alors nous pouvons trouver l'optimum de $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ comme $\tilde{G}_{\mathcal{B},\eta}^{\mathsf{OSTS}\star} = \min_\lambda \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$. De plus, on peut montrer que lorsque nous augmentons λ, tous $d_{k,\lambda}$ vont diminuer. Par conséquent, la valeur maximale de λ est λ^{max} satisfaisant $\mathcal{H}_0(\omega_k^{\mathsf{f}}, d_{k,\lambda^{\mathsf{max}}}) \ge f_k^{\mathsf{f,rq}}$, $\forall k \in \mathcal{B}$ and $\sum_{k\in\mathcal{B}} d_{k,\lambda^{\mathsf{max}}} \le D^{\mathsf{max}}$.

Notez que nous pouvons arrêter le processus de recherche lorsqu'il existe un λ tel que $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \le F^{\mathsf{f,max}}$. Lorsque la recherche de bissection de η converge, nous pouvons trouver l'optimum $\lambda^\star = \mathrm{argmin}_\lambda\, \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$, et les variables optimales $s_k^{\mathsf{m}\star} = s_{k,\lambda^\star}^{\mathsf{m}}$, $s_k^{\mathsf{f}\star} = s_{k,\lambda^\star}^{\mathsf{f}}$, $s_k^{\mathsf{c}\star} = 1 - s_k^{\mathsf{m}\star} - s_k^{\mathsf{f}\star}$, $f_k^{\mathsf{f}\star} = s_{k,\lambda^\star}^{\mathsf{m}} f_{k,\lambda^\star} + s_{k,\lambda^\star}^{\mathsf{f}} f_k^{\mathsf{f,rq}}$, and $d_k^\star = s_{k,\lambda^\star}^{\mathsf{m}} d_{k,\lambda^\star} + (1 - s_{k,\lambda^\star}^{\mathsf{f}} - s_{k,\lambda^\star}^{\mathsf{m}}) d_k^{\mathsf{rq}}, \forall k \in \mathcal{B}$. L'algorithme OSTS pour la vérification de faisabilité de $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ est résumé dans Algorithm 2.7.

b2) Mise à Jour itérative de λ basée sur un algorithme en deux étapes (IUTS Alg.) Cette méthode peut résoudre $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ avec une très faible complexité via les mises à jour du dual lagrangien. Plus précisément, la fonction duale de $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ peut être définie comme $\mathcal{G}^{\mathsf{o}}(\lambda) = \min_{\tilde{\Omega}_4} \mathcal{L}(\tilde{\Omega}_4, \lambda)$, et le problème dual peut être défini comme

$$
\max_\lambda \mathcal{G}^{\mathsf{o}}(\lambda) \quad \text{s. t.} \quad \lambda \ge 0. \tag{2.36}
$$

Comme le problème dual est toujours convexe, $\mathcal{G}^{\mathsf{o}}(\lambda)$ peut être maximisé en utilisant la méthode du sous-gradient dans laquelle la variable dual λ est mise à jour de manière itérative comme suit:

$\lambda_n = \left[\lambda_{n-1} + \delta_n\Big(\sum_{k\in\mathcal{B}}\left(s^{\mathsf{m}}_{k,\lambda_{n-1}} d_{k,\lambda_{n-1}} + s^{\mathsf{c}}_{k,\lambda_{n-1}} d^{\mathsf{rq}}_k\right) - D^{\mathsf{max}}\Big)\right]^+$, où n désigne le numéro d'itération, δ_n représente la taille du déplacement, et $[a]^+$ est défini comme $\max(0, a)$. La méthode du sous-gradient est garantie de converger vers la valeur optimale de λ pour un point initial initial Ω_4 si la taille du déplacement, δ_n est choisie de manière appropriée, par exemple, $\delta_n \to 0$ lorsque $n \to \infty$, qui est respectée par en définissant $\delta_n = 1/\sqrt{n}$. Pour un λ_n donné, nous pouvons déterminer la variable primaire $d_{k,\lambda_n} = \text{inv}(\mathcal{H}_2(\lambda_n))$. Pour λ_n et d_{k,λ_n} donné, le problème primal devient un programme linéaire dans $s_{k,\lambda_n}, \forall k \in \mathcal{B}$, qui peut être résolu efficacement en utilisant des techniques d'optimisation linéaire standard. De plus, les sommets de ce problème sont les points où les $s^{\mathsf{m}}_{k,\lambda_n}$, $s^{\mathsf{f}}_{k,\lambda_n}$, et $s^{\mathsf{c}}_{k,\lambda_n}$ sont soit 0 soit 1. Ainsi, *résoudre le problème relaxé renverra également des valeurs binaires* 0 *ou* 1.

Cependant, étant donné les valeurs 0 ou 1 des variables $s^{\mathsf{m}}_{k,\lambda_n}$, $s^{\mathsf{f}}_{k,\lambda_n}$, la décision sur l'emplacement d'exécution de l'application (brouillard ou nuage) peut être bloquée dans une solution optimale locale de sorte que les ressources de calcul de brouillard requises ne peuvent pas être mises à jour pour améliorer la solution. Pour surmonter ce problème critique, la méthode du gradient projeté peut être adoptée pour mettre à jour lentement les variables $s^{\mathsf{m}}_{k,\lambda_n}$, $s^{\mathsf{f}}_{k,\lambda_n}$, et $s^{\mathsf{c}}_{k,\lambda_n}$ comme $\mathbf{s}_k^{(n+1)} = \mathbb{P}_{\Phi_k}\left(\mathbf{s}_k^{(n)} - \check{\delta}\nabla\mathbf{s}_k^{(n)}\right)$, où $\mathbf{s}_k^{(n)} = [s^{\mathsf{m}}_{k,\lambda_n}, s^{\mathsf{f}}_{k,\lambda_n}, s^{\mathsf{c}}_{k,\lambda_n}]$, $\check{\delta}$ est la taille du déplacement, $\nabla\mathbf{s}_k^{(n)} = [\mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_{k,\lambda_n}) + \lambda_n d_{k,\lambda_n}, \lambda_n f^{\mathsf{f,rq}}_k, \lambda_n d^{\mathsf{rq}}_k]$ et $\mathbb{P}_{\Phi_k}(.)$ est la projection sur l'ensemble $\Phi_k = \Big\{\mathbf{s}_k | \mathbf{s}_k \geq 0,\ s^{\mathsf{f}}_{k,\lambda_n} + s^{\mathsf{c}}_{k,\lambda_n} + s^{\mathsf{m}}_{k,\lambda_n} \leq 1\Big\}$. Enfin, on peut vérifier que ce mécanisme itératif converge toujours [30].

2.2.2.4 Résultats Numériques

Nous considérons un système informatique de brouillard-nuage hiérarchique composé de $K = 10$ utilisateurs répartis au hasard dans la zone de couverture cellulaire avec un rayon de 800 m et la BS est située au centre de la cellule. En particulier, l'affaiblissement de propagation est calculé comme $\beta_k(\text{dB}) = 128.1 + 37.6\log_{10}(dist_k)$, où $dist_k$ est la distance géographique entre l'utilisateur k et le BS (en km) [27]. Nous avons en outre défini le gain de formation de faisceau comme $M_0 = 5$, la bande passante de transmission maximale comme $\rho^{\mathsf{max}}_k = 1$ MHz et la densité de puissance de bruit comme $\sigma_{\mathsf{bs}} = 1.381 \times 10^{-23} \times 290 \times 10^{0.9}$ W/Hz [31]. Tous les utilisateurs sont supposés avoir la même vitesse d'horloge maximale de $2,4$ GHz, une puissance d'émission maximale de $p^{\mathsf{max}}_k = 0.22$ W et la consommation électrique du circuit par Hz est définie comme $p_{k,0} = 22$ nW/Hz. Nous supposons

que le nombre de bits de transmission nécessaires pour prendre en charge le déchargement de calcul b_k^{in} est le même pour tous les utilisateurs. De plus, les demandes de calcul des 10 utilisateurs $\{c_1, c_2, ..., c_9, c_{10}\}$ sont définies de manière aléatoire dans la plage $1.8 - 2.4$ Gcycles tandis que le délai maximal est $T_k^{\mathsf{max}} = 1$ seconde, la charge non déchargeable est $c_{k,0} = 0.1c_k$ et la charge déchargeable est $c_{k,1} = 0.9c_k$ pour tous les utilisateurs.

Nous avons également défini le coefficient d'énergie comme $\alpha_k = 0.1 \times 10^{-27}$ et le temps de calcul sur le serveur de nuage comme $T^{\mathsf{c}} = T_k^{\mathsf{max}}/5$. Pour l'algorithme de compression de données, nous définissons $\gamma_{k,1}^{\mathsf{co}} = 0.03 \times 2.6^{32.28}$, $\gamma_{k,2}^{\mathsf{co}} = 32.28$, $\gamma_{k,3}^{\mathsf{co}} = 0.3$, $\gamma_{k,1}^{\mathsf{de}} = 0.115$, $\gamma_{k,2}^{\mathsf{de}} = -0.9179$, $\gamma_{k,3}^{\mathsf{de}} = 0.046, \forall k$, $\omega_k^{\mathsf{u,min}} = 2.3$, et $\omega_k^{\mathsf{u,max}} = 2.9$. Les poids d'énergie et de délai sont choisis de telle sorte que $w_k^{\mathsf{E}} + w_k^{\mathsf{T}} = 1, \forall k$. Les résultats de la simulation sont obtenus en faisant la moyenne de plus de 100 réalisations des emplacements aléatoires des utilisateurs. Enfin, pour toutes les figures, nous définissons la taille des données brutes comme $b_k^{\mathsf{in}} = 4$ Mbits (sauf pour la Fig. 2.3), $w_k^{\mathsf{E}} = 2w_k^{\mathsf{T}}$, $\forall k$, la ressource de calcul de brouillard maximale en $F^{\mathsf{f,max}} = 15$ GHz, la capacité d'amenée maximale en $D^{\mathsf{max}} = 20$ Mbps et $\kappa = 50$ (sauf pour les Figs. 2.3, où κ capture la relation entre $\gamma_{k,0}^{\mathsf{u}}$ in (2.20) et la taille des données brutes est $\gamma_{k,0}^{\mathsf{u}} = \kappa b_k^{\mathsf{in}}$ [32]. En pratique, un serveur de brouillard peut prendre en charge des algorithmes de compression de données plus puissants que ceux des utilisateurs. Cela implique que le ratio de compression du serveur de brouillard est beaucoup plus élevé que celui des utilisateurs. Par conséquent, lorsque le serveur de brouillard décompresse et recompresse les données, nous définissons les paramètres comme suit: $\gamma_{k,1}^{\mathsf{co,f}} = 0.076, \gamma_{k,2}^{\mathsf{co,f}} = 0.7116, \gamma_{k,3}^{\mathsf{co,f}} = 0.5794$, $\omega_k^{\mathsf{f,min}} = 3.4$, et $\omega_k^{\mathsf{f,max}} = 11.2$. La taille du déplacement, est définie comme $\check{\delta} = 0.1$.

Dans la Fig. 2.3, nous montrons les avantages significatifs de la compression de données pour le déchargement de calcul où le WEDC min-max (appelé WEDC pour plus de concision) vs b_k^{in} est tracé pour six schémas différents: le schéma 'Local-execution' dans lequel toutes les applications des utilisateurs sont exécutées localement; l'Alg. in [15] (w/o Comp)' dans lequel l'algorithme de référence dans [15] est utilisé avec $\omega_k^{\mathsf{u}} = 1, \forall k$ et pas de compression de données; le 'JCORA Alg. w/o Comp' dans lequel l'algorithme JCORA est utilisé avec $\omega_k^{\mathsf{u}} = 1, \forall k$; et pas de compression de données (les autres variables sont optimisées comme dans l'algorithme JCORA); et trois autres instances de l'algorithme JCORA avec la compression de données et trois valeurs différentes de $\kappa = 50, 100, 200$ ($\kappa = \gamma_{k,0}^{\mathsf{u}}/b_k^{\mathsf{in}}$). Pour garantir une comparaison équitable entre les 'Alg. in [15] (w/o Comp)' et nos schémas, nous appliquons également MIMO et optimisons la décision de déchargement et l'allocation des ressources de calcul du brouillard, la puissance de transmission, la bande passante, et la vitesse

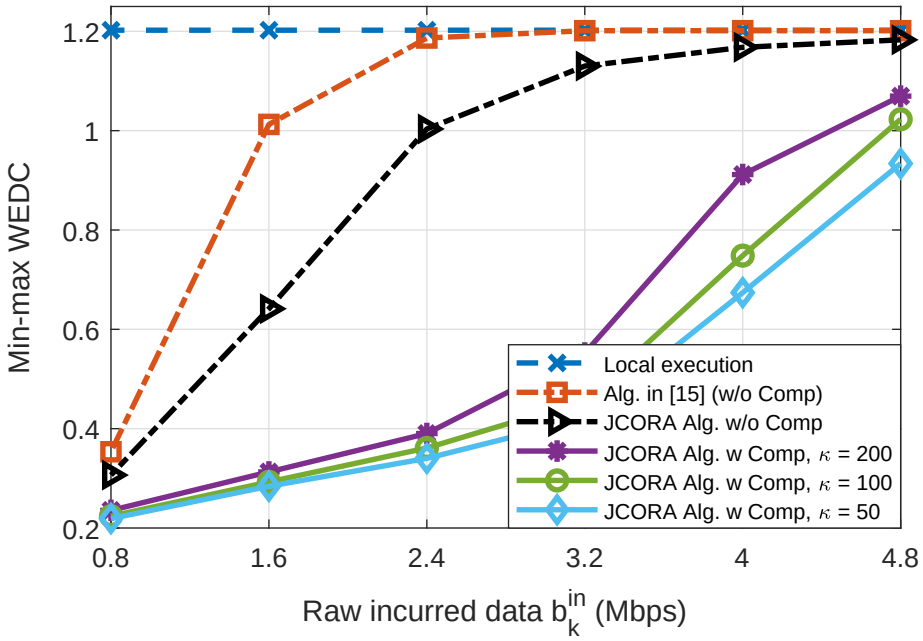

Figure 2.3 – Min-max WEDC vs. b_k^{in}.

d'horloge du processeur local pour l'Alg. in [15] (w/o Comp)'. De plus, pour la variable restante d_k, nous allouons la capacité d'amenée également aux utilisateurs qui déchargent leurs tâches sur le serveur de nuage.

Comme on peut le voir sur la Fig. 2.3, le déchargement des calculs peut grandement améliorer le WEDC lorsque les ressources radio et calcul sont suffisantes pour prendre en charge le déchargement (par exemple, la quantité de données requise n'est pas trop importante). Plus précisément, le déchargement de calcul même sans de compression de données peut entraîner une réduction significative du WEDC par rapport à l'exécution locale, en particulier lorsque la quantité de données encourues b_k^{in} est petite, de sorte que les ressources radio limitées ne limitent pas performance. De plus, même sans exploiter la compression des données, notre proposition (JCORA Alg. w/o Comp) donne de bien meilleures performances que l'algorithme proposé dans [15]. En effet, notre conception proposée optimise conjointement les décisions de déchargement et l'allocation des ressources informatiques et radio, tandis que dans [15], les décisions de déchargement se trouvent presque indépendantes de l'allocation des ressources de calcul et radio. En particulier, la technique de relaxation semi-définie utilisée dans [15] peut ne pas toujours garantir la condition de rang-1 pour la matrice optimisée. L'optimisation conjointe de la compression de données, du déchargement des calculs, et de l'allocation des ressources peut conduire à une réduction du WEDC pour une plus grande plage de b_k^{in} (par exemple, lorsque $b_k^{in} = 2,4$ Mbps, le WEDC min-max est réduit jusqu'à 65

%). Cependant, l'énergie et le temps consommés pour la (dé) compression affectent également le WEDC min-max réalisable, et leur impact a tendance à devenir plus fort pour les plus grands $\gamma_{k,0}^{\mathsf{u}}$ et lorsque le les ressources radio disponibles sont plus limitées.

Pour évaluer les performances du système lorsque la compression de données est effectué à la fois par les utilisateurs mobiles et par le serveur de brouillard, nous considérons le paramétrage suivant: $\gamma_{k,0}^{\mathsf{f}} = \gamma_{k,0}^{\mathsf{u}}$, $F^{\mathsf{f,max}} = 15$ GHz, et $D^{\mathsf{max}} = 20$ Mbps. Les avantages de la recompression des données dans le brouillard sont illustrés à la Fig. 2.4 où nous traçons le WEDC min-max en fonction de b_k^{in} pour quatre schémas différents: le 'JCORA Alg. w Schéma Comp 'dans lequel les données sont compressées uniquement par les utilisateurs tandis que les trois schémas restants correspondent aux algorithmes proposés pour le cas étendu. En particulier, les labels '9-pt PLA Alg. w Fog Comp', 'OSTS Alg. w Fog Comp', et 'IUTS Alg. w Fog Comp' correspondent aux algorithmes PLA, OSTS, et IUTS à 9 points, respectivement, qui effectuent la compression à la fois pour les utilisateurs et le serveur de brouillard. Pour $b_k^{\mathsf{in}} = 4$ Mbits, une réduction WEDC min-max supplémentaire de 35 % peut être obtenue en effectuant un compression de données sur les utilisateurs et le serveur de brouillard. De plus, les ressources radio requises diminuent avec la diminution de b_k^{in}; par conséquent, le gain est réduit en raison de la baisse de la demande de transmission de données. Lorsque b_k^{in} augmente, le principal goulot d'étranglement pour le déchargement des calculs sont les ressources radio limitées disponibles pour prendre en charge les transmissions de données entre les utilisateurs et le serveur de brouillard; par conséquent, le gain dû à la recompression des données sur le serveur de brouillard devient moins important. Ce chiffre confirme également que les schémas '9-pt PLA', 'OSTS', et 'IUTS' atteignent presque le même WEDC min-max.

2.2.3 Planification sans Fil de Services Hétérogènes avec Numérologie Mixte dans les Réseaux sans Fil 5G

Les principales contributions de ce travail de recherche peuvent être résumées comme suit:

- Nous étudions le problème de planification de services hétérogènes à numérologie mixte qui vise à maximiser le nombre d'utilisateurs admis tout en répondant aux exigences de latence des services et de transmission de données.

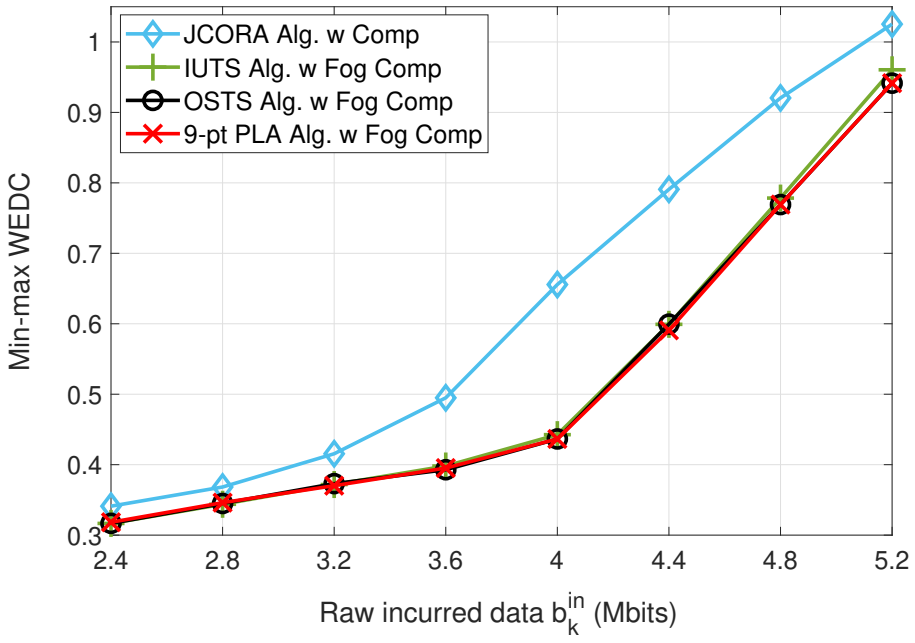

Figure 2.4 – Min-max WEDC in general design scenario.

- Nous proposons deux algorithmes, nommés algorithme basé sur le partitionnement des ressources (RPA) et algorithme gourmand itératif (IGA), pour acquérir des solutions efficaces de planification des ressources.

2.2.3.1 Modèle de Système

Nous considérons le système 5G où la ressource temps-fréquence disponible est divisée en éléments de ressource (RE). Chaque RE occupe la bande passante de $\Delta^{\mathsf{f}}_{\mathsf{min}}$ (Hz) et la durée du slot de $\Delta^{\mathsf{t}}_{\mathsf{min}}$ (secondes). Les conditions de liaison/canal pour chaque sous-porteuse sont supposées inchangées pendant le temps de planification. De plus, nous supposons que l'allocation des ressources est effectuée sur chaque intervalle de planification de $T = N^{\mathsf{t}}\Delta^{\mathsf{t}}_{\mathsf{min}}$ (secondes) et la bande passante de $B^{\mathsf{f}} = M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}$ (Hz). Considérant le problème de planification pour les utilisateurs où la BS prend en charge plusieurs numérologies. La bande passante d'un PRB en numérologie l est définie comme Δ^{f}_l et la durée du slot d'un PRB en numérologie l est définie comme Δ^{t}_l où $\Delta^{\mathsf{t}}_l = \Delta^{\mathsf{t}}_{l-1}/2$, $\Delta^{\mathsf{f}}_l = 2\Delta^{\mathsf{f}}_{l-1}$, $\Delta^{\mathsf{t}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{t}}_l, \forall l\}$, et $\Delta^{\mathsf{f}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{f}}_l, \forall l\}$. Pour des raisons de commodité, la numérologie utilisée par l'utilisateur k est désignée par l_k, l'ensemble de tous les utilisateurs est indiqué par $\mathcal{K}$, l'ensemble des numérologies est indiqué par $\mathcal{L}$, $l_{\mathsf{max}} = \max\{l \in \mathcal{L}\}$, $l_{\mathsf{min}} = \min\{l \in \mathcal{L}\}$, et $L = l_{\mathsf{max}} - l_{\mathsf{min}}$ et les cardinaux de l'ensemble $\mathcal{L}$ sont désignés par $|\mathcal{L}|$. Chaque utilisateur k

nécessite qu'un bloc de données de d_k^{rq} bits soit entièrement transmis et le temps d'attente total pour sa transmission de données ne doit pas être supérieur à τ_k^{max}. Nous utilisons (i, j) pour faire référence à un RE particulier où son emplacement est donné comme $f \in [(i-1)\Delta_{\mathsf{min}}^{\mathsf{f}} : i\Delta_{\mathsf{min}}^{\mathsf{f}})$ and $t \in [(j-1)\Delta_{\mathsf{min}}^{\mathsf{t}} : j\Delta_{\mathsf{min}}^{\mathsf{t}})$, for $1 \leq i \leq M^{\mathsf{f}}$ and $1 \leq j \leq N^{\mathsf{t}}$.

2.2.3.2 Formulation du Problème

Les PRB sont attribués aux utilisateurs dont la numérologie est sélectionnée à l'avance par chaque utilisateur. De plus, chaque RE est allouée à un seul utilisateur et associée à une numérologie. Nous représentons la correspondance d'un PRB particulier de numérologie l aux RE dans l'espace de ressources fréquence-temps 2D comme suit:

$$\boldsymbol{q}_{l,m,n} = \{(i,j) | m \leq i \leq m + M_l, n \leq j \leq n + N_l\}, \tag{2.37}$$

où $M_l = 2^{l-l_{\mathsf{min}}} - 1$, $N_l = 2^{l_{\mathsf{max}}-l} - 1$, en supposant que le nombre de PRB attribués à l'utilisateur k n'est pas supérieur à C_k pour maintenir une certaine équité entre les utilisateurs. Ensuite, nous introduisons les variables binaires $x_{i,j}^{k,c}$, $y_{m,n}^{k,c}$ où $y_{m,n}^{k,c} = 1$ si $\boldsymbol{q}_{l_k,m,n}$ correspond au c^{th} PRB affecté de l'utilisateur k et $y_{m,n}^{k,c} = 0$ sinon; $x_{i,j}^{k,c} = 1$ si RE (i, j) est affecté à l'utilisateur k dans son c^{th} PRB et $x_{i,j}^{k,c} = 0$ sinon. Pour $k \in \mathcal{K}$, les plages de c, i, j, m et n sont $c = 1 : C_k$, $i = 1 : M^{\mathsf{f}}$, $j = 1 : N^{\mathsf{t}}$, $m = 1 : M^{\mathsf{f}} - M_{l_k}$, et $n = 1 : N^{\mathsf{t}} - N_{l_k}$, respectivement. Nous imposons les contraintes suivantes pour garantir une allocation des ressources sans chevauchement:

$$\sum_{i'=m:m+M_{l_k}} \sum_{j'=n:n+N_{l_k}} x_{i',j'}^{k,c} \geq 2^L y_{m,n}^{k,c}, \ \forall k, c, m, n, \tag{2.38a}$$

$$\sum\nolimits_{k\in\mathcal{K}} \sum\nolimits_{c} x_{i,j}^{k,c} \leq 1, \forall i, j, \text{ and } \sum\nolimits_{m} \sum\nolimits_{n} y_{m,n}^{k,c} \leq 1, \forall k, c. \tag{2.38b}$$

Soit $r_{m,n}^k$ le débit de transmission de l'utilisateur k sur PRB $\boldsymbol{q}_{l_k,m,n}$. Ensuite, la quantité totale de données transmises par l'utilisateur k pendant l'intervalle de planification est $d_k = \Delta_{l_k}^{\mathsf{t}} \sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}} \sum_{n=1}^{N^{\mathsf{t}}-N_{l_k}} \sum_{c=1}^{C_k} r_{m,n}^k y_{m,n}^{k,c}$. Chaque utilisateur k souhaite que son bloc de données soit entièrement transmis et que le temps d'attente total ne soit pas supérieur à τ_k^{max}. Soit $\tau_{k,0}$ le temps d'attente initial (du bloc de données) de l'utilisateur $k \in \mathcal{K}$ au début de l'intervalle de

planification considéré. [2] Ensuite, le temps d'attente total jusqu'à l'instant de transmission de l'utilisateur k peut être écrit comme $\tau_k = \tau_{k,0} + \tau_{k,1}$, où $\tau_{k,1}$ est le temps d'attente supplémentaire avant que l'utilisateur k ne soit servi dans l'intervalle de planification, qui peut être exprimé comme $\tau_{k,1} = \Delta^{\mathsf{t}}_{\mathsf{min}} \min\{j-1 \mid x^{k,c}_{i,j}=1, \forall i,j,c\}$.

Notre conception vise à planifier autant d'utilisateurs que possible tout en répondant à leurs exigences de demande de données et de latence. Rappelons que l'utilisateur k souhaite transmettre un bloc de données de d^{rq}_k bits avec le temps d'attente total pas plus grand que τ^{max}_k. Pour maintenir ces contraintes, nous définissons une fonction de capture si les deux contraintes sont satisfaites par $u_k = \mathbb{1}_{d_k - d^{\mathsf{rq}}_k} \mathbb{1}_{\tau^{\mathsf{max}}_k - \tau_k}$, où $\mathbb{1}_x$ représente la fonction step, c'est-à-dire $\mathbb{1}_x = 1$ si $x \geq 0$ et $\mathbb{1}_x = 0$, sinon. En fait, si une solution d'ordonnancement garantit que la quantité de données transmises et le temps d'attente total satisfont $d_k \geq d^{\mathsf{rq}}_k$ and $\tau_k \leq \tau^{\mathsf{max}}_k$, respectivement, nous avons $u_k = 1$; sinon, $u_k = 0$. Ensuite, le problème de planification peut être formulé comme

$$(\mathcal{P}_1) \max_{\boldsymbol{x},\boldsymbol{y}} \sum\nolimits_{k\in\mathcal{K}} u_k \text{ s.t. } (2.38a), (2.38b), \text{ and } \boldsymbol{x}, \boldsymbol{y} \in \{0,1\},$$

where $\boldsymbol{x} = \{x^{k,c}_{i,j} | \ \forall i,j,k,c\}$ and $\boldsymbol{y} = \{y^{k,c}_{m,n} | \ \forall k,c,m,n\}$.

Soit $z^{k,c}_{m,n} = u_k y^{k,c}_{m,n}, \forall k,c,m,n$, $(\mathcal{P}_1)$ peut être transformé en formulaire ILP comme

$$
\begin{aligned}
(\mathcal{P}_1^{\mathsf{ILP}}) \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \ \sum_{k\in\mathcal{K}} u_k \\
& \text{s. t.} \quad (2.38a),\ (2.38b), \\
& \sum_{m} \sum_{n'=1:N^{\mathsf{rq}}_k} \sum_{c} z^{k,c}_{m,n'} - u_k \geq 0,\ \forall k \in \mathcal{K}, && \text{(2.39a)} \\
& \Delta^{\mathsf{t}}_{l_k} \sum_{m}\sum_{n}\sum_{c} r^{k}_{m,n} z^{k,c}_{m,n} - u_k d^{\mathsf{rq}}_k \geq 0,\ \forall k \in \mathcal{K}, && \text{(2.39b)} \\
& z^{k,c}_{m,n} \geq u_k + y^{k,c}_{m,n} - 1,\ \ z^{k,c}_{m,n} \leq \min\{u_k, y^{k,c}_{m,n}\},\ \forall k,c,m,n, && \text{(2.39c)}
\end{aligned}
$$

où $\boldsymbol{z} = \{z^{k,c}_{m,n}, z^{k,c}_{m,n} | \ \forall k,c,m,n\}$ et $\boldsymbol{u} = \{u^{\mathsf{r}}_k, u^{\mathsf{d}}_k | \ \forall k\}$. Ici, N^{rq}_k représente le nombre maximal d'intervalles de temps (avec une taille de $\Delta^{\mathsf{t}}_{\mathsf{min}}$ secondes) que l'utilisateur k peut attendre, en comp-

[2] Ce temps d'attente initial est appliqué au premier bloc d'un nouveau flux de données lorsque le flux de données arrive au milieu de l'intervalle de planification précédent.

Algorithm 2.8. Resource Partitioning based Algorithm (RPA)

1: Initialize: Set initial value for M_{B}.
2: Partition resources into M_{B} sub-bands and distribute users into these sub-bands as in **Section 2.2.3.3**.
3: **Step 1**: Solve $(\mathcal{P}^{\mathsf{ILP}}_{1,m_{\mathsf{B}}})$ to obtain $u_k^{\mathsf{S1}\star}$ for all m_{B}, $k \in \mathcal{K}_{m_{\mathsf{B}}}$.
4: **Step 2**: Solve $(\mathcal{P}_{m_{\mathsf{B}}})$ to create a contiguous region of unallocated resources between two consecutive sub-bands while still satisfying the requirements of admitted users, i.e., users with $u_k^{\mathsf{S1}\star} = 1$, $\forall k$.
5: **Step 3**: Solve $(\mathcal{P}_{\mathsf{RPA}})$ to assign unallocated resources to un-admitted users, i.e., users with $u_k^{\mathsf{S1}\star} = 0$, $\forall k$.

Algorithm 2.9. Iterative Greedy Algorithm (IGA)

1: Initialize: $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_k, c_k = 0, \mathcal{W}_{m,n} = 1, W = 10$.
2: **repeat**
3: Compute $\mathcal{U}_{m,n,k}$ find the largest value of $\mathcal{U}_{m_0,n_0,k_0}$, and perform the corresponding PRB allocation.
4: Update different parameters after the PRB allocation as $c_{k_0} = c_{k_0} + 1$, asign $y^{k_0,c_{k_0}}_{m_0,n_0} = 1$, update the remaining required data $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_{k,0} - r^{k_0}_{m_0,n_0}\Delta^{\mathsf{t}}_{l_k}$, and $\mathcal{W}_{m_0,n} = W, \forall n = 1 : N^{\mathsf{t}}$.
5: Drop all overlapped PRBs $\boldsymbol{q}_{l_k,m,n}$ to PRB $\boldsymbol{q}_{l_{k_0},m_0,n_0}$
6: **until** $\mathcal{U}_{m,n,k} = 0, \forall m, n, k$

tant à partir de le début de l'intervalle de planification, pour respecter sa contrainte de délai qui est déterminée comme $N^{\mathsf{rq}}_k = \lfloor(\tau^{\mathsf{max}}_k - \tau_{k_0})/\Delta^{\mathsf{t}}_{\mathsf{min}}\rfloor$, où $\lfloor.\rfloor$ est la partie entière.

2.2.3.3 Algorithmes Proposés

a) Algorithme basé sur le Partitionnement des Ressources (RPA)

Nous proposons un algorithme de faible complexité qui résout $(\mathcal{P}^{\mathsf{ILP}}_1)$ en le décomposant en sous-problèmes parallèles à petite échelle. Nous divisons d'abord la bande passante disponible en M_{B} sous-bandes où la sous-bande m_{B} occupe le spectre de

$$(m_{\mathsf{B}} - 1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}} \text{ à } \min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}}, M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}\}$$

Ensuite, les trois étapes suivantes sont suivies dans RPA: 1) effectuer l'allocation des ressources sur chaque sous-bande, 2) réorganiser les ressources non allouées pour les sous-bandes consécutives, et 3) affecter ces ressources réorganisées pour en prendre en charge plus d'utilisateurs.

Les étapes clés de RPA sont résumées dans **Algorithm** 2.8. Dans **Step 1**, nous distribuons aléatoirement les utilisateurs en sous-bandes pour rendre les demandes de ressources sur différentes

sous-bandes similaires. Dénoter l'ensemble des utilisateurs associés à la sous-bande m_{B} comme $\mathcal{K}_{m_{\mathsf{B}}}$, et l'ensemble des RE dans la dimension de fréquence comme $\mathcal{I}_{m_{\mathsf{B}}} = \{m | m = (m_{\mathsf{B}}-1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor : \min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor, M^{\mathsf{f}}\}\}$. On résout alors $(\mathcal{P}_1^{\mathsf{ILP}})$ correspondant à chaque sous-bande m_{B} et l'ensemble d'utilisateurs $\mathcal{K}_{m_{\mathsf{B}}}$ pour obtenir les décisions d'admission, notées $u_k^{\mathsf{S1}\star}$. Le sous-problème de la sous-bande m_{B} est nommé $(\mathcal{P}_{1,m_{\mathsf{B}}}^{\mathsf{ILP}})$. Dans **Step 2**, après avoir trouvé les $u_k^{\mathsf{S1}\star}$, nous réarrangons les ressources allouées afin que les RE non alloués de deux sous-bandes consécutives puissent être disposés à proximité l'un de l'autre et ils peuvent être combinés et mappés dans des PRB d'une certaine numérologie comme défini dans (2.37). Le réarrangement des RE non alloués sur la sous-bande m_{B} peut être obtenu en résolvant le problème suivant:

$$\begin{aligned}(\mathcal{P}_{m_{\mathsf{B}}}) \quad & \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in\{0,1\}} \sum_{k\in\mathcal{K}_{m_{\mathsf{B}}}} \sum_{i\in\mathcal{I}_{m_{\mathsf{B}}}} \sum_{j=1:N^{\mathsf{t}}-N_{l_k}} \sum_{c=1:C_k} W_i x_{i,j}^{k,c} \\ \text{s.t.} \quad & u_k{=}u_k^{\mathsf{S1}\star}, \quad (2.38a),(2.38b),(2.39a),(2.39b),\ \forall k{\in}\mathcal{K}_{m_{\mathsf{B}}}, i{\in}\mathcal{I}_{m_{\mathsf{B}}},\end{aligned}$$

où $\{W_i\}$ est une série croissante, par exemple, $W_i{=}2^i$ si l'indice de sous-bande est impair et $\{W_i\}$ est une série décroissante, par exemple, $W_i{=}2^{-i}$ si l'indice de sous-bande est pair. On peut vérifier qu'après avoir résolu $(\mathcal{P}_{m_{\mathsf{B}}})$, tous les RE non alloués dans deux sous-bandes consécutives seront poussés l'un près de l'autre pour créer une région de ressources contiguës aussi grande que possible. Soit $\{\boldsymbol{x}_{\mathcal{P}_{m_{\mathsf{B}}}}^\star, \boldsymbol{y}_{\mathcal{P}_{m_{\mathsf{B}}}}^\star, \boldsymbol{z}_{\mathcal{P}_{m_{\mathsf{B}}}}^\star\}$ désigne la solution optimale de $(\mathcal{P}_{m_{\mathsf{B}}})$. Dans **Étape 3**, nous affectons les ressources non allouées, $\Omega = \{(i,j) | \boldsymbol{x}_{\mathcal{P}_{m_{\mathsf{B}}}}^\star = 0, \forall m_{\mathsf{B}}\}$, à l'ensemble des utilisateurs non admis $\bar{\mathcal{K}} = \{k | u_k^{\mathsf{S1}\star} = 0\}$en résolvant le problème suivant $(\mathcal{P}_{\mathsf{RPA}})$:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \sum_{k\in\bar{\mathcal{K}}} u_k \text{ s.t. } (2.38a),(2.38b),(2.39a),(2.39b), \forall k{\in}\bar{\mathcal{K}}, (i,j){\in}\Omega.$$

b) Iterative Greedy Algorithm (IGA) - Algorithme gourmand itératif

Nous proposons un autre algorithme itératif rapide dans lequel nous attribuons avidement des ressources aux utilisateurs en fonction d'un poids d'affectation qui dépend des exigences de l'utilisateur sous-jacent, de la quantité de données transmises et de la latence obtenue si le PRB sous-jacent est attribué à l'utilisateur. Dans chaque itération, nous calculons le poids d'affectation pour chaque paire d'un PRB disponible et un utilisateur sur la base duquel l'affectation des ressources est effectuée pour la paire d'utilisateur PRB atteignant le poids le plus élevé. Après cela, les PRB

disponibles et les poids de toutes les paires d'utilisateurs PRB possibles sont mis à jour pour préparer une nouvelle affectation de ressources dans la prochaine itération. Ce processus est répété jusqu'à ce qu'il n'y ait plus de PRB disponible ou d'utilisateur insatisfait. Nous définissons maintenant le poids d'affectation des ressources pour une paire particulière PRB-utilisateur-ressource comme suit: $\mathcal{U}^k_{m,n} = \frac{r^k_{m,n}\Delta^{\mathsf{t}}_{l_k}}{d^{\mathsf{rq}}_{k,0}} \frac{n}{N^{\mathsf{rq}}_k} \mathbb{1}_{n\in\mathcal{N}_k}\mathcal{W}_{m,n}$ if $c_k \leq C_k, d^{\mathsf{rq}}_{k,0} > 0$, and $\mathcal{U}^k_{m,n} = 0$, sinon, où $d^{\mathsf{rq}}_{k,0}$ est la quantité de données requise restant dans chaque itération, qui est égal à d^{rq}_k dans la première itération, c_k est le total actuel des PRB attribués à l'utilisateur k et $\mathcal{N}_k$ est l'ensemble des RE dans le domaine temporel, qui est défini comme $\mathcal{N}_k = \{n|n \leq N^{\mathsf{rq}}_k\}$ if $\sum_{n=1}^{N^{\mathsf{rq}}_k}\sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}}\sum_{c=1}^{C_k} y^{k,c}_{m,n} = 0$, et $\mathcal{N}_k = \{n|n \leq N^{\mathsf{t}}\}$ sinon, et $\mathcal{W}_{m,n}$ est une matrice utilisée pour atténuer la fragmentation des ressources dans le processus d'allocation, qui est mis à jour à chaque itération. La matrice unitaire est initialement affectée à $\mathcal{W}_{m,n}$.

En particulier, $\mathcal{U}^k_{m,n}$ est choisi en fonction des critères suivants: 1) Les utilisateurs avec un bloc de données requis plus petit reçoivent des priorités de planification plus élevées; 2) Si un utilisateur n'est pas encore admis, un PRB à l'emplacement temporel le plus proche de la tranche correspondant au temps d'attente maximum autorisé est plus prioritaire; 3) les utilisateurs admis se voient allouer des ressources jusqu'à ce que leurs besoins soient entièrement satisfaits; et 4) la fragmentation des ressources est empêchée pour faciliter les allocations futures du PRB. L'IGA est résumé dans **Algorithm 2.9**. Dans chaque itération d'IGA, nous devons calculer les poids d'affectation des ressources $\mathcal{U}_{m,n,k}$ pour tous les m, n, k disponibles, déterminer le plus grand $\mathcal{U}_{m_0,n_0,k_0}$ pour effectuer une RA, mettre à jour différents paramètres et déposer tous les PRB superposés dans le bloc affecté. La complexité la plus défavorable de chaque itération est $\mathcal{O}(M^{\mathsf{f}}N^{\mathsf{t}}K)$. Soit N^{iter} le nombre d'itérations, qui est délimité par $M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|$, le pire des cas la complexité de l'IGA est $\mathcal{O}(N^{\mathsf{iter}}M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|K)$.

2.2.3.4 Résultats Numériques

Nous considérons un système sans fil avec des piétons et des utilisateurs très mobiles dans une cellule d'un rayon de 500 mètres. L'affaiblissement de propagation de canal β_k (dB) $= 128.1 + 37.6\log_{10}(\gamma_k)$ où γ_k est la distance entre l'utilisateur k et le BS (en km). Pour les évanouissements de canaux à petite échelle, les paramètres du canal piéton B de l'ITU avec un décalage Doppler de 50 Hz et les paramètres du canal Véhicule-A de l'ITU avec un décalage Doppler de 500 Hz sont utilisés

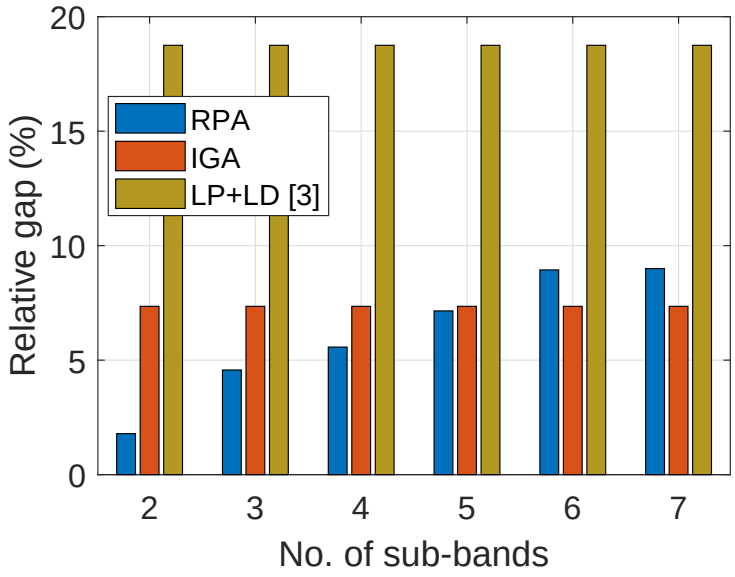

Figure 2.5 – Comparison of RPA and IGA with the optimum on the relative gap

respectivement pour les utilisateurs piétons et les utilisateurs en mouvement. Nous avons défini trois groupes d'utilisateurs A, B et C et le nombre d'utilisateurs dans ce groupe est $\lfloor K/3 \rfloor$, $\lfloor K/3 \rfloor$ et $K-2\lfloor K/3 \rfloor$, respectivement. Plus précisément, le groupe A adopte la numérologie 0 correspondant aux utilisateurs en mouvement élevé avec une grande demande de données, le groupe C utilise la numérologie 2 correspondant aux utilisateurs en mouvement élevé nécessitant un temps d'attente faible, et le groupe B utilise la numérologie 1 correspondant aux utilisateurs piétons ayant des exigences moyennes sur les données de transmission et temps d'attente.

Le débit de transmission est calculé en fonction de la capacité de Shannon où le rapport de la puissance de transmission par Hz à la densité de puissance de bruit est fixé à 2.8×10^5. Les blocs de données requis d_k^{rq} sur l'intervalle T de 1 ms pour les utilisateurs des groupes A, B et C sont définis de manière aléatoire dans $[500-2000]$ (bits), $[500-1000]$ (bits) et $[180-500]$ (bits), respectivement, et C_k est défini égal à 10. De plus, N_k^{rq} défini dans la proposition 1 est défini égal à 8 pour les utilisateurs du groupe A et au hasard dans [3-6] et [1-4] pour les utilisateurs des groupes B et C, respectivement. Tous les résultats numériques sont obtenus en faisant la moyenne des résultats sur 50 réalisations aléatoires.

Nous montrons l'écart relatif dans la Fig. 2.5 qui est calculé comme $(\sum_k u_k^{\mathsf{G}} - \sum_k u_k^{\mathsf{RPA/IGA}}) \times 100\%/\sum_k u_k^{\mathsf{G}}$ for $M^{\mathsf{f}} = 32$ et $K = 20$ où $u_k^{\mathsf{RPA/IGA/LP+LD[3]}}$ et u_k^{G} représentent les valeurs de la fonction objectif pour l'utilisateur k obtenues en utilisant RPA/IGA/LP+LD[4] et le solveur CVX-Gurobi, respectivement. La Fig. 2.6 montre le rapport entre le temps d'exécution moyen (ET) du

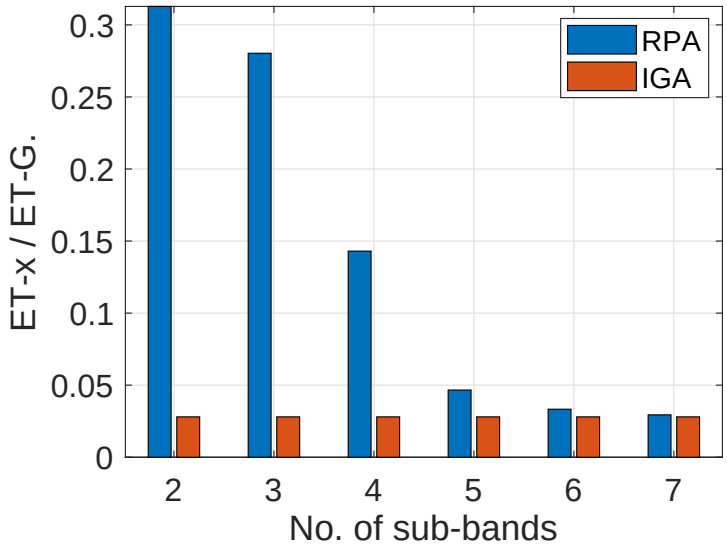

Figure 2.6 – Comparison of RPA and IGA with the optimum on the execution time

RPA / IGA et celui requis pour résoudre ($\mathcal{P}_1^{\mathsf{ILP}}$) par le solveur CVX-Gurobi (ET-G). Nous avons choisi u_k^{G} comme solution référencée car l'écart relatif entre la solution optimale réelle et la solution renvoyée par le solveur Gurobi est inférieur à 10^{-4} par défaut. L'algorithme "LP + LD" proposé dans [4] comprend deux boucles: une boucle externe pour attribuer des PRB aux utilisateurs en fonction de la matrice d'utilité pour toutes les paires d'utilisateurs PRB, et une boucle interne pour déterminer la matrice d'utilité. Plus précisément, la matrice d'utilité est déterminée en considérant la relaxation de programmation linéaire (LP) et le problème lagrangien dual (LD).

En fait, l'algorithme "LP + LD" obtient sa solution en s'attaquant au problème duale ILP où on ne peut garantir qu'il n'y aura pas de saut de dualité. La figure montre que nos algorithmes surpassent l'algorithme "LP + LD". De plus, comme le montre cette figure, l'écart relatif dû au RPA augmente lorsque le nombre de sous-bandes M_{B} augmente. En effet, une plus grande somme de M_{B} réduit la flexibilité d'allocation de resources dans chaque sous-bande et donc l'efficacité d'utilisation des ressources. Cependant, le temps d'exécution du RPA peut être considérablement réduit lorsque M_{B} devient plus grand. En revanche, IGA explore toujours de bonnes paires ressources-utilisateurs pour une utilisation efficace des ressources et IGA n'est pas affecté par M_{B}.

2.3 Remarques Finales

Dans cette thèse de doctorat, nous avons développé diverses nouvelles techniques de gestion des ressources et algorithmes pour les systèmes exploitant 5G NR et MEC. En particulier, nous avons apporté trois contributions importantes à la recherche. Premièrement, nous avons développé des algorithmes de décision d'allocation et de déchargement des ressources écoénergétiques pour MEC, qui fonctionnent bien mieux que les autres stratégies de calcul locales conventionnelles en termes d'économie d'énergie et d'équité. Ensuite, nous avons proposé une énergie pondérée efficace et un coût de délai général pour les systèmes informatiques de brouillard-nuage hiérarchiques via la compression des données, la décision de déchargement et l'allocation des ressources conjointe. Les conceptions proposées surpassent considérablement les autres conceptions de pointe de la littérature. Enfin, nous étudions le problème de plainification de services hétérogènes à numérologie mixte qui vise à maximiser le nombre d'utilisateurs admis tout en répondant aux exigences de latence de service et de transmission de données et démontrons l'efficacité des algorithmes proposés pour déterminer les solutions.

Chapter 3

Introduction

3.1 Background and Motivation

The research on the fifth-generation (5G) and beyond wireless cellular networks has been driven by the need to support the mobile traffic explosion and the rapidly increasing number of wireless communications, including both human-based and Internet-of-Things (IoT) connections. Specifically, it is predicted that tens of billions of wireless devices, from low-cost IoT to smartphones, tablets, virtual reality headsets, and cars will be connected to wireless networks over the next few years [2]. The communication demand on different kinds of mobile devices in vertical domains including the smart factory, smart vehicles, smart grid, smart city is more and more sophisticated. Thus, future wireless networks must provide different communication services with different QoS requirements. In particular, International Telecommunication Union (ITU) classifies 5G mobile network services into three categories: enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC), and ultra-Reliable and Low Latency Communications (uRLLC). In general, eMBB refers to bandwidth-hungry services, such as high-definition video and virtual/augmented reality (VR/AR) streamings. As such, mMTC is suitable in scenarios with dense connectivity, such as smart cities and smart farming. This is in distinction to uRLLC, which is aimed at supporting the time-sensitive networking and mission-critical services, such as automatic/assisted driving and remote control. To provide these services, a new frame structure has been defined in 5G New Radio (NR). By allowing the flexible numerology, the Transmission Time Interval (TTI) and frame-size can be flexibly configured to fit with the service demands [3]. Some early works [4–6] on the scheduling

and resource allocation with 5G NR frame structure show a promising way to enhance the future system performance.

Besides, an important aspect of 5G wireless network is the application's view. Indeed, with recent breakthroughs in artificial intelligence (AI), new emerging applications enabling new ways of interactions among things and humans have been created to enhance the quality of life. Many of them are compute-intensive applications such as e-health, object recognition/detection/monitoring. When only communications-related issues are concerned in network design and management, it is impossible to enable these compute-intensive applications on many different kinds of devices, especially low-cost IoT devices. Therefore, 5G wireless networks must support not only communication, but also computation, control, and content delivery (4C) functions. Mobile Edge Computing (MEC) has been recently proposed as an important technology in 5G wireless networks to enable a variety of new compute-intensive applications even on low-cost IoT devices. In general, MEC is a network architecture concept defined by ETSI [20], that enables cloud computing capabilities and an IT service environment at the edge of the cellular network. Different design aspects of MEC, such as task partitioning and resource allocation, have been investigated in both academic and industry communities to enable them and support future system scenarios and applications [21, 22]. One typical network architecture for the 5G wireless system is shown in Fig. 3.1, which employs various enabling technologies and novel network architectures for efficient support of various wireless applications.

3.1.1 From Mobile Cloud to Mobile Edge Computing and its Variants

In general, mobile edge/cloud computing (MEC/MCC) technologies enable enhancing the mobile usability and prolonging the mobile battery life by offloading computation-intensive applications to a remote fog/cloud server [7–9]. In an MCC system, enormous computing resources are available in the core network, but the limited backhaul capacity can induce significant delay for the underlying applications. In contrast, an MEC system, with computing resources being deployed at the network edge in close proximity to the mobile devices, can enable computation offloading and meet demanding application requirements [10]. Hierarchical fog-cloud computing systems which leverage the advantages of both MCC and MEC can further enhance the system performance [11–15] where fog servers deployed at the network edge can operate collaboratively with the more powerful cloud

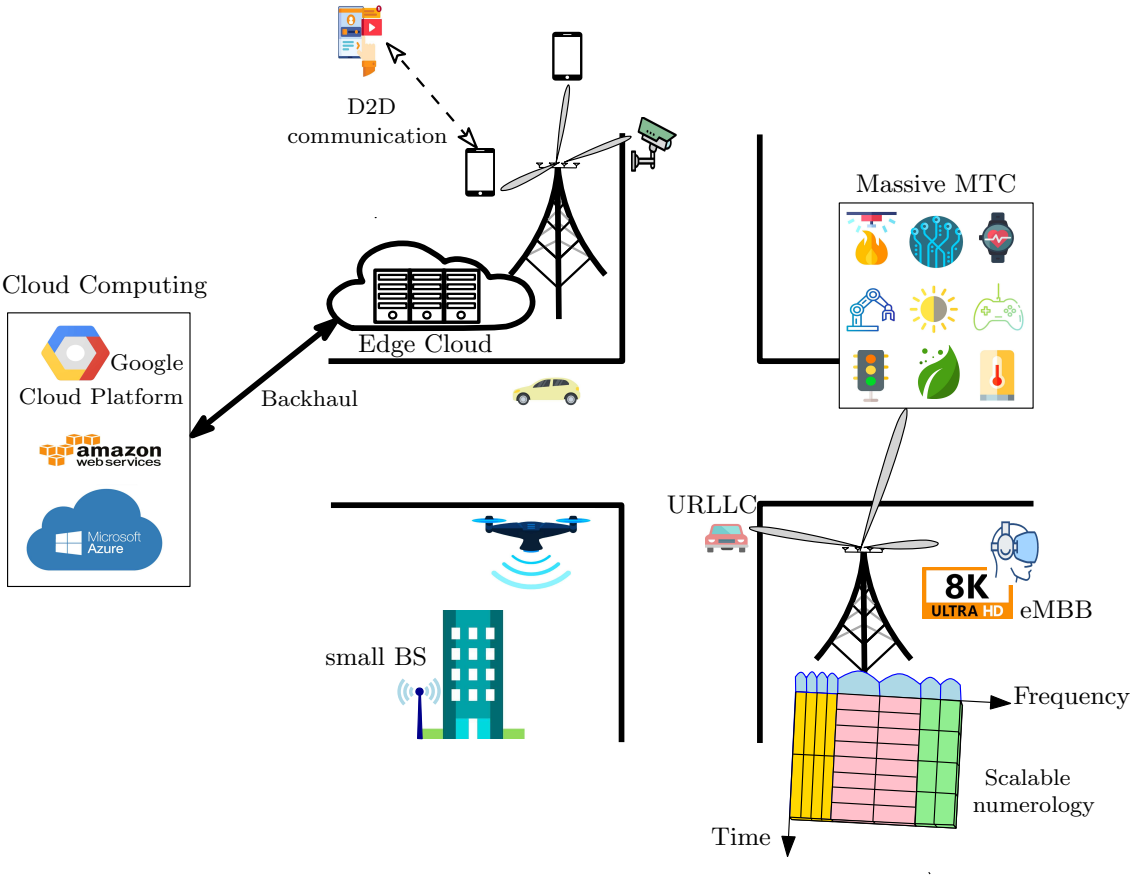

Figure 3.1 – 5G system exploiting edge/cloud computing and New radio technology.

servers to execute computation-intensive user applications. Specifically, when the users' applications require high computing power or low latency, their computation tasks can be offloaded and processed at the fog and/or remote cloud servers. The potential scenarios and applications of MEC and its variants are still being discussed for 5G and beyond systems.

Due to the need for data exchange incurred in the offloading process, the wireless transmission plays an integral role in the computational offloading system [16]. Therefore, to efficiently utilize MEC power, one needs to develop efficient designs for joint management of both wireless and computing resources. Moreover, advanced communications technologies such as massive MIMO, heterogeneous network (HetNet), and device-to-device (D2D), which allow enhancing the spectral efficiency will be employed in MEC and its variants. Accordingly, the joint management of two different kinds of resources becomes very challenging. Specifically, different from the general wireless networks, the energy and time delay in MEC and its variants are related not only to wireless transmission but also to computation factors. They are complicated functions of different parameters and factors such as bandwidth, transmit power, CPU clock speed, execution location.

Effective management and optimization of these parameters in two different kinds of resources is a very challenging problem[17], and still requires significantly more concerted effort from the wireless community [18, 19].

3.1.2 Wireless Communication Services Enabled by 5G New Radio

5G NR is an entirely new air interface being developed for 5G to support a wide variety of services and devices. This part will briefly introduce a new concept in 5G NR, named numerology. Note that numerology is a term which is used to define the grid of discrete resources in the continuous time-frequency plane. A critical feature in 5G NR is the utilization of carrier frequency from sub-1 GHz up to 52.6 GHz, as defined in the 3rd Generation Partnership Project (3GPP) Release 15. However, the channel propagation properties of low and high frequency bands are very different. In particular, the low frequency band is strongly affected by the delay spread intensive environments while the low frequency band is strongly influenced by the phase noise [23]. Accordingly, a single numerology applied for a wide range of frequencies becomes inefficient or even impossible. Comparing to LTE numerology, 5G NR supports multiple types of subcarrier spacing based on a baseline subcarrier spacing of 15 kHz [24]. In particular, 5G NR defines five distinct OFDM numerologies which is parameterized as μ, $\mu = 0, 1, 2, 3, 4$ as given in Table 3.1. The numerology μ has the subcarrier bandwidth of $2^{\mu} \times 15$ kHz and the slot duration of $2^{-\mu}$ milliseconds. In numerology $\mu = 0$, the time-frequency grid is the same with LTE, 5G NR can be coexistence with LTE and the LTE-based NB-IoT on the same subcarrier. For the lower frequency bands with narrow subcarrier spacing, numerologies 0, 1, and 2 are used to counter the delay spread intensive environments. For the higher frequency bands, using numerology 2, 3, and 4 with wide subcarrier spacing can make the system robust to the phase noise, and can support low latency services efficiently [24]. The introduction of different numerologies provides the flexibility for various services in the same system. It also introduces new challenges when multiplexing different numerologies in the same time-frequency space. Due to the different time-frequency grid, the wireless scheduling for heterogeneous services with mixed numerology in the 5G wireless networks becomes an important problem to improve the system performance.

Computation and radio resource management for MEC systems and wireless scheduling for heterogeneous services considering NR mixed numerology are research problems considered in the

Parameters	Numerology μ				
	0	1	2	3	4
Subcarrier spacing (kHz)	15	30	60	120	240
Slot duration (ms)	1	0.5	0.25	0.125	0.0625
OFDM symbol duration (μs)	66.67	33.33	16.67	8.33	4.17

Table 3.1 – Numerology structure in 5G

dissertation. To this end, we consider three fundamental design aspects to allow the above coexistence, namely resource allocation and offloading decision in MEC systems, joint data compression, offloading decision, and resource allocation in hierarchial fog-cloud systems, and wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks. In the following, we discuss research challenges related to these design problems.

3.2 Research Challenges

For complicated systems exploiting MEC or 5G NR, they require sophisticated resource management design to enable robust and efficient coexistence of multiple kinds of services. However, various research challenges arise as one considers different system models, communications and computation settings, design objectives, and constraints. We discuss some of these significant challenges in the following subsections.

3.2.1 Resource Allocation and Offloading Decision in MEC Systems

Energy-efficient resource allocation problem is formulated to enable efficient MEC system operation and utilization of wireless and computation resources. However, there remain many technical challenges.

First, in the offloading process, it needs to efficiently decide whether the computation task is locally executed or offloaded and remotely processed. This becomes very challenging when the computation task cannot be partitioned and must be executed entirely either in the local end-device or in a remote cloud server. Indeed, the ofloading decisions in this case are the binary variables, and the optimization problems related to binary/integer variables are difficult to tackle in general.

Second, the overall energy and time delay include two terms: one related to the wireless transmission and one related to the computation process. Accordingly, to make the system robust and

achieve high performance, it needs to control both computation-related and wireless-related parameters. In general, it much more challenging to manage compared to control only wireless-related parameters.

The final challenge concerns joint offloading decision and resource allocation, which involves a mixed-integer optimization problem. Moreover, the channel gains of individual links are different, and the computation workloads of users can be very different. Therefore, the designed joint offloading decision and resource allocation algorithms must intelligently determine the appropriate subtasks for offloading and the proper resources to support the offloading process to achieve the best performance.

3.2.2 Offloading Decision and Resource Allocation in Hierarchial Fog-Cloud Systems

Hierarchical fog-cloud computing systems that leverage the advantages of both MCC and MEC can further enhance the system performance. However, besides the challenges encountered in MEC systems, we have to deal with further difficulties in hierarchical fog-cloud computing systems. The first challenge is related to data reduction when the transmission time from each user to the cloud becomes a major factor and directly affects the Quality of Service (QoS). In fact, the lack of an appropriate model for capturing data compression in the existing literature creates difficulty when designing the hierarchical fog-cloud computing systems. To have the insightful results, it needs to build a suitable model to express the relation between the compression/decompression computation workloads and the compression ratio.

The second challenge is related to the system's modeling. In particular, hierarchical fog-cloud computing systems involve in two communication links which are mobile user - fog and fog - cloud links. The communication conditions are different between these two types of communication. In particular, the wireless links are used between the mobile users and the fogs, while the wired links are usually employed between the fogs and the cloud. The QoS constraints on computation in fog and cloud are also different. Therefore, a proper system model is needed to capture the key characteristics of the hierarchical fog-cloud computing systems.

The third challenge concerns the joint data compression, computation offloading, and resource allocation in hierarchical fog-cloud systems. Compared to MEC systems, the energy and time delay in this system are much more complicated because of data compression-related parameters and two-tier communication and computing architecture. Moreover, the data compression makes the resource-relation among users more sophisticated. Therefore, to achieve a high system performance, the proposed algorithm must address the strong coupling among users.

3.2.3 Wireless Scheduling for Hetrogeneous Services with Mixed Numerology in 5G Wireless Networks

To guarantee the required quality of service (QoS) and achieve efficient resource utilization, one must determine the set of users to be scheduled over resource blocks since the wireless network may not be able to support all users concurrently. For the 5G wireles systems with mixed numerology, there exists many different types of physical resource blocks (PRBs), hence, the joint scheduling and PRB assignments is a vital design. Toward this end, we have to deal with various challenges, as described in the following. Firstly, one should appropriately represent the mapping for one particular PRB in the frequency-time resource space, and capture the multiplexing of different numerologies in the same time-frequency space. Secondly, we need to model the scheduling problem with data demand and QoS requirements. Thirdly, solving the wireless scheduling for heterogeneous services with mixed numerology problems can be very challenging because these problems are typically in the form of integer programming (IP) or integer linear programming (ILP), which is NP-Hard in general. Therefore, low-complexity algorithms must be proposed to tackle the scheduling problem and achieve acceptable results when compared to optimal or close-to-optimal algorithms.

3.3 Literature Review

In the following, we survey the existing literature related to our research studies. Firstly, we describe the related works on energy-oriented and delay-oriented offloading designs for MEC systems. Secondly, we present the computation offloading researches for different task models in MEC. Thirdly, the papers on resource allocation and offloading decision in MEC are summarized. Finally, we review the recent papers on applying the numerology in 5G NR for system performance enhancement.

3.3.1 Energy-oriented and Delay-oriented Computation Offloading in MEC System

Computation offloading design for MCC/MCE systems has been studied extensively in the literature, see recent surveys [22, 33] and the references therein. Most existing works consider one or both of the two main performance metrics in their designs, namely energy-efficiency maximization [34–37] and delay-efficiency minimization [38–41]. Focusing on energy-efficiency, the authors of [34] develop partial offloading frameworks for multiuser MEC systems employing time division multiple access (TDMA) and frequency-division multiple access (FDMA). In [35], wireless power transfer is jointly designed with the computation offloading design. Moreover, different binary offloading frameworks are developed in [36, 37] where branch-and-bound and heuristic algorithms are proposed to solve the resulting mixed-integer optimization problems.

Considering the delay performance, an iterative heuristic algorithm to optimize the binary offloading decisions for minimization of the total computation and transmission delay in a hierarchical fog-cloud system is proposed in [38]. The authors in [39] formulate the computation offloading and resource allocation problem as a student-project-allocation game to maximize the ratio between the average offloaded data rate and the offloading cost at the users. In [40], the authors study a binary computation offloading problem for maximization of the weighted sum computation rate. Then, they propose a coordinate descent based algorithm in which the offloading decision and time-sharing variables are iteratively updated until convergence. Considering partial computation offloading and a TDMA based resource sharing strategy, the authors in [41] propose a framework for minimization of the weighted-sum latency of the mobile users for the collaborative cloud and fog computing system.

Some recently proposed schemes for computation offloading consider both energy and delay performance metrics [13, 15, 16]. In particular, the work in [13] proposes a radio and computing resource allocation framework, where the computational loads of the fog and cloud servers are determined to achieve the desirable trade-off between power consumption and service delay. Additionally, the authors of [16] jointly optimize the transmit power and offloading probability for minimization of the average weighted energy, delay, and payment cost. In [15], the authors study fair computation offloading design that minimizes the maximum weighted cost of delay and energy consumption among all users in a hierarchical fog-cloud system. In this work, a two-stage algorithm is proposed

where the offloading decisions are determined in the first stage using a semidefinite relaxation and probability rounding based method while the radio and computing resource allocation is determined in the second stage. However, references [13, 15, 16, 34–40] have not exploited data compression for computation offloading.

3.3.2 Computation Offloading Designs for Different Scenarios

In the existing literature, computation offloading problems are studied in four important scenarios: single-task multi-user systems, single-task multi-user systems, multi-task single-user systems, and multi-task multi-user systems. Computation offloading for the multi-task single-user system is considered in [42–44]. In particular, the task offloading is dynamically optimized by using the Lyapunov optimization method in [42] or by solving a constrained shortest path problem in [43]. In [44], the authors formulate the link selection and data transmission scheduling as a discrete-time stochastic dynamic program to optimize the energy efficiency.

For the single-task multi-user system, the computation offloading and resource allocation optimization becomes more complicated [15, 34, 45–47]. Partial computation offloading to minimize the weighted sum of energy consumption is studied in [34], where CPU partitioning decision variables are assumed to be continuous. In [15, 45, 48], the authors consider the single-task offloading and resource allocation for the interference-free transmission scenario. In [45], the computing and communication resources are assumed to be shared among users with the control assistance of the network operator. This proposed framework exploits available network resources to achieve the required communication delay and high energy-efficiency. In [15], the authors consider the mixed fog/cloud architecture and leverage the computing resources both at the fog and the cloud while the work [48] employs the Lyapunov-based optimization technique to minimize the average weighted sum power consumption where the computation tasks of mobile users are assumed to be fine-grained. Moreover, iterative algorithms are developed in [46, 47] to optimize the energy-efficiency considering radio interference. Specifically, the decomposition framework and heuristic algorithm are respectively proposed in [46] and [47] to deal with the complicated intra- and inter-cell interference. However, making offloading decisions based on certain priority in [47] may not guarantee fairness among mobile users.

While the above papers consider three different MEC systems, namely the single-task single-user system, single-task multi-user system, and multi-task single-user system, the more complex multi-task multi-user system has been studied in some recent works [49–51]. Specifically, for given the wireless transmission rate, the work [49] proposes a heuristic algorithm to tackle the offloading and task scheduling problem for multicore-based mobile devices. Considering the interference-free wireless network, binary task offloading design is conducted in [50] by employing semi-definite relaxation and the probability based rounding technique. However, both transmit energy and spectral efficiency are assumed to be unchanged in bandwidth allocation optimization in this work, which may not hold true in practice. This optimization framework is further extended in [51] to consider co-channel interference in the heterogeneous network based MEC system; however, independent power allocation for different users adopted by this work may not be efficient in managing the interference. Even though there have been some efforts in tackling the computation offloading design for the multi-task multi-user MEC system, consideration of advanced communication aspects such as MIMO communications for such design requires much more further research.

3.3.3 Resource Allocation and Offloading Design in MEC Systems

In order to guarantee the service latency requirements and successful transmission of the involved data to the cloud in an energy-efficient manner, it is vital to jointly optimize the computation and radio resources [52]. Along this line, Zhang et al. in [28] consider probabilistic computation offloading and study an optimal strategy to minimize the energy consumption assuming wireless transmission over a Gilbert-Elliott channel. Aiming to maximize the revenue of mobile virtual network operators (MVNOs), the authors of [53] formulate the virtual resource allocation as a joint optimization problem; however, the authors employ the relaxation technique to solve the problem that might not provide any performance guarantee.

Considering the wireless powered MEC and TDMA protocol, the work [54] introduces the energy-effective resource allocation policy in a two-user system. Along the way, the paper [35] develops a partial offloading framework for multi-user system that jointly optimizes the transmit energy at access points, users' transmit power, and users' CPU clock speed for delay constrained energy minimization problem. The work [55] employs the Nash bargaining game to formulate the fair resource allocation problem that aims to maximize the overall network throughput. However,

the constraint on the positive transmission rates for all users makes the computation offloading optimization. Coalition game is utilized in [56] to tackle the total weighted delay cost minimization problem. Moreover, the Lyapunov optimization technique is applied in [48, 57, 58] to minimize users' power consumption while trading off resources allocated for local computation and task offloading. In [38], the offloading decision is optimized for latency-sensitive applications. The transmission rate-related parameters, offloading decision, and users' CPU clock speed are optimized in [59] to minimize the local overhead and in [36] to minimize the total energy consumption of all users in the system. In addition, computation offloading is optimized for energy-saving in the hierarchical MEC system in [60, 61]. Considering the reliability, the work [62] formulates an optimization problem to jointly minimize the latency and offloading failure probability.

Existing resource allocation designs mostly consider simple settings such as the scenarios with only two users in the system [54], uncontrolled wireless resource [60, 61], or simple constraint [55]. Moreover, heuristic and sub-optimal algorithms developed in several existing works may not provide strong performance guarantees with respect to the optimal solution [36, 62] and they may have very high computational complexity [35].

Despite these research progresses, joint resource allocation and computation offloading has attracted a lot of research interest. In particular, innovative wireless techniques, new architectures, and advanced design tools have been employed in different design scenarios such as multiple-input multiple output (MIMO) nonorthogonal multiple access (NOMA)-MEC [63], device-to-device (D2D)-MEC [62, 64], blockchain-MEC [65, 66], millimeter wave (mmWave)-MEC [67], unmanned aerial vehicle (UAV)-MEC [68–75], inter-user task dependency [76], deep reinforcement learning [77–80].

3.3.4 Enabling Techniques to Support Heterogeneous Services with 5G NR

5G NR provides standardized framework for the 5G wireless network; however, research required to realize the benefits of 5G NR is still ongoing. The very first works studying the performance benefits of wireless systems leveraging different mini-slot sizes given by 5G NR size can be seen in [81–84]. In particular, various simulations are performed to evaluate the performance of grant-free uRLLC and eMBB multiplexing in uplink in [81] while the work [82] selects the suitable TTI and allocates the channels to support multi-services simultaneously. In [83], the authors develop a dynamic time

division duplex (TDD) based scheduling framework for a scenario with mixed types of services while the paper [84] perform the performance analysis of orthogonal and non-orthogonal multiple access for the multiplexing of eMBB and uRLLC users using the Shannon based rate bound for the finite blocklength packet. However, these works [81, 83, 84] only consider different mini-slot sizes and neglect the difference in subcarrier spacing in different numerologies. In [85], they investigate a metric to capture the average user satisfaction considering inter-numerology interference, spectral efficiency reduction, complexity, and signaling overhead based on which a heuristic algorithm is proposed to decide on the number of mixed numerologies used in the same system. Meanwhile, the papers [6, 86] investigate the interference pattern in mixed and single numerology spectrum sharing systems. In particular, the work [86] analyzes the intercarrier interference and intersymbol interference for different values of subcarrier spacings considering imperfect channel estimation while the paper [87] proposes an interference balancing transceiver that aims to reduce the variance of the interference energy caused by non-orthorgonal multiplexing of different numerologies. By assuming that each eMBB user is assigned only one PRB for data transmission, the works [3, 88] derive the outage probability and transmission rate for different multiplexing strategies enabling the coexistence of uRLLC and eMBB services. The resource allocation of two-dimensional (2D) time-frequency resources for maximizing the cloud-radio access network (C-RAN) operator's revenue is studied in [89]. However, the authors of [3, 88, 89] assume that only two different numerologies exist in the system.

Few existing works explore resource allocation for system performance enhancement when considering full aspects of of the 5G mixed numerology. Specifically, the authors of [4, 5] study the resource allocation problem with mixed numerology for capacity enhancement. In [6], a resource allocation problem is formulated to support multiple numerologies simultaneously where the design objective is to achieve fairness among service flows. There are still many open research issues which must be addressed in order to realize the benefits of 5G NR.

3.4 Research Objectives and Contributions

The general objective of my Ph.D research is to develop efficient resource management techniques to leverage the MEC and 5G NR to enable heterogeneous wireless services and applications in future wireless systems. Specifically, our main contributions can be described as follows.

1. Energy-efficient resource allocation and computation offloading designs for MEC systems:

   We formulate the computation task offloading and resource allocation optimization in multiple-input multiple-output (MIMO) based MEC systems considering perfect/imperfect-channel state information (CSI) estimation that aims to minimize the maximum weighted energy consumption subject to practical constraints on available computing and radio resources and allowable latency. The optimal and low-complexity algorithms are proposed to solve the underlying mixedd interger non-linear programming (MINLP) problems. For the perfect-CSI, we employ bisection search and develop a novel mechanism to find the optimal solution. The low-complexity algorithms are developed by decomposing the original optimization problem into the offloading optimization (OP) and power allocation (PA) subproblems and solve them iteratively. Moreover, the difference of convex functions (DC) method is employed to deal with non-convex structure of (PA) in the imperfect-CSI scenario. We show the advantages of proposed designs over conventional local computation strategies in terms of energy-saving and fairness.

2. Joint data compression, computation offloading and resource allocation for hierarchical fog-cloud systems:

   We propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by data compression and decompression. We then formulate the joint optimization problem of the compression ratio, computation offloading, and resource allocation to minimize the maximum weighted energy and service delay cost (WEDC) of all users considering data compression at only the mobile users and at both the mobile users and the fog server. First, when data compression is performed only at the mobile users, we prove that the optimal offloading decisions have a threshold structure. Moreover, a novel three-step approach employing convexification techniques is developed to optimize the compression ratios and the resource allocation. Then, we address the more general design where data compression is performed at both the mobile users and the fog server. We propose three algorithms to overcome the strong coupling between the offloading decisions and the resource allocation. We show that our proposed designs outperform conventional computation offloading strategies that do not leverage data compression or use sub-optimal optimization approaches.

3. Wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks:

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. 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. Because the complexity of solving an integer linear programming (ILP) problem can be decreased exponentially if we decrease the number of optimization variables, the RPA algorithm is developed by partitioning the resources and users into smaller groups, optimizing the RA for each group, then performing resource defragmentation and additional resource assignment for unallocated resources to obtain a final solution. In the IGA algorithm, we iteratively allocate PRBs to users based on an appropriate resource assignment weight to obtain an efficient scheduling solution with low computation complexity. Finally, extensive numerical studies are performed to demonstrate the efficacy of the proposed algorithms.

3.5 Dissertation Outline

The remaining of this dissertation is organized as follows. Chapter 4 reviews some fundamental mathematical background and important models used in the dissertation. In Chapter 5, we present the energy-efficient resource allocation and computation offloading design for MEC systems. Then, we study the joint data compression, computation offloading and resource allocation design for hierarchical fog-cloud systems in Chapter 6. In Chapter 7, we describe wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks. Chapter 8 summarizes the main contributions of the dissertation and makes some recommendations for future research.

Chapter 4

Background

This chapter presents some fundamentals of optimization and resource allocation for the MEC system. In particular, various optimization techniques which are utilized to model and solve various resource allocation problems in this dissertation are also presented.

4.1 Mathematical Optimization

4.1.1 Basic Terminology

A mathematical optimization problem can be described as follows [1]:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s. t.} \quad & f_i(\boldsymbol{x}) \leq 0,\ i = 1, ..., m \\ & h_i(\boldsymbol{x}) = 0,\ i = 1, ..., p, \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the optimization variable and $f_0(\boldsymbol{x})$ is an objective function. The set of $\boldsymbol{x}$ that satisfies all m inequality and p equality constraints is called the feasible set. If the feasible set is empty, the problem is infeasible. A vector $\boldsymbol{x}^*$ is called optimal of the problem if it achieves the smallest objective value among all optimization variables $\boldsymbol{x}$ satisfying the constraints.

4.1.2 Convex Optimization

Convex optimization is a well studied sub-field in optimization theory. A fundamental property of convex optimization problems is that any locally optimal point is also (globally) optimal, thus it is easier to solve compared to a general optimization problem. In the following, some fundamentals of convex optimization are briefly introduced.

4.1.2.1 Definition [1]

Convex set: A set S is convex if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y} \in S. \tag{4.2}$$

Convex function: A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if the domain $\mathcal{D}$ of f is a convex set and f satisfies the following condition

$$f(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y}). \tag{4.3}$$

Convex optimization problem: An optimization problem is convex if it has the following form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s. t.} \quad & f_i(\boldsymbol{x}) \leq 0, \ i = 1, ..., m \\ & h_i(\boldsymbol{x}) = 0, \ i = 1, ..., p, \end{aligned} \tag{4.4}$$

where the function $f_0, f_1, ..., f_m$ are convex and the function $h_1, h_2, ..., h_p$ are linear.

The Lagrangian: The Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ of problem (4.4) has the following form

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i h_i(\boldsymbol{x}), \tag{4.5}$$

where vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are called the dual variables or Lagrange multiplier vectors associated with the inequality and equality constraints of problem (4.4), respectively.

The Lagrangian dual function: The Lagrangian dual function $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as follows

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x} \in \mathcal{D}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x} \in \mathcal{D}} \left(f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i h_i(\boldsymbol{x}) \right). \tag{4.6}$$

The Lagrangian dual problem: The Lagrangian dual problem associated with problem (4.4) is defined as follows

$$\begin{aligned} \max_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \quad & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{s. t.} \quad & \boldsymbol{\lambda} \succeq 0. \end{aligned} \tag{4.7}$$

Problem (4.4) is so-called the primal problem. Let p^* and d^* denote the optimal values of problem (4.4) and problem (4.7), respectively. The important inequality $d^* \leq p^*$ holds for any optimization problem, and this property is called weak duality. The difference $p^* - d^*$ is refereed as the optimal duality gap. When $p^* = d^*$, the optimal duality gap is zero, and the problem has strong duality.

4.1.2.2 Karush-Kuhn-Tucker (KKT) Conditions

Let $\boldsymbol{x}^*$ and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ be the primal and dual optimal solutions under the strong duality condition. Then, they will satisfy the KKT conditions which are expressed as follows

$$\begin{aligned} & f_i(\boldsymbol{x}^*) \leq 0, \ \forall i = 1, ..., m \\ & h_i(\boldsymbol{x}^*) = 0, \ \forall i = 1, ..., p \\ & \lambda_i^* \geq 0, \ \forall i = 1, ..., m \\ & \lambda_i^* f_i(\boldsymbol{x}^*) = 0, \ \forall i = 1, ..., m \\ & \nabla f_0(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(\boldsymbol{x}^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(\boldsymbol{x}^*) = 0, \end{aligned} \tag{4.8}$$

where ∇f_i is the gradient of function f_i.

The KKT conditions are sufficient conditions to obtain an optimal solution for a convex optimization problem. It means that any points $\boldsymbol{x}^*$ and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ satisfying the KKT conditions are the primal and dual optimal solutions.

4.1.3 Geometric Programming

Convex optimization problems can be solved by several well-known optimization algorithms such as the interior point method, ellipsoid method, subgradient projection method [1]. However, most optimization problems are non-convex by their nature. Geometric program represents a family of optimization problems that are not convex, however, they can be transformed into convex optimization problems, by a change of variables and a transformation of the objective and constraint functions.

4.1.3.1 Definition

Monomial function: A function $f : \mathbb{R}^n \to \mathbb{R}$ with $\mathrm{dom} f = \mathbb{R}^n_{++}$, defined as

$$f(\boldsymbol{x}) = c \prod_i x_i^{a_i}, \tag{4.9}$$

where $c > 0$ and $a_i \in \mathbb{R}$, is called a monomial function, or simply, a monomial.

Posynomial function: A posynomial function is the sum of monomials, defined as

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_i x_i^{a_{ik}}, \tag{4.10}$$

where $c_k > 0$ and $a_{ik} \in \mathbb{R}$.

Geometric program: An optimization problem is a geometric program if it has the following form

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n_{++}} \quad & f_0(\boldsymbol{x}) \\ \text{s. t.} \quad & f_i(\boldsymbol{x}) \leq 0, \ i = 1, ..., m \\ & h_i(\boldsymbol{x}) = 0, \ i = 1, ..., p, \end{aligned} \tag{4.11}$$

where the function $f_0, f_1, ..., f_m$ are posynomials and the function $h_1, h_2, ..., h_p$ are monomials.

4.1.3.2 A Convex Form of Geometric Program

Let $y_i = \log(x_i)$, so $x_i = \exp(y_i)$. The monomial $f(\boldsymbol{x}) = c\prod_i x_i^{a_i}$ can be transformed as $f(\boldsymbol{y}) = \exp(\boldsymbol{a}^T\boldsymbol{y} + b)$, where $b = \log(c)$. Then, the equivalent geometric program (4.11) expressed in terms of new variable $\boldsymbol{y}$ can be stated as

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \sum_{k=1}^{K_0} \exp({\boldsymbol{a_{0k}}}^T\boldsymbol{y} + b_{0k}) \\ \text{s. t.} \quad & \sum_{k=1}^{K_i} \exp({\boldsymbol{a_{ik}}}^T\boldsymbol{y} + b_{ik}) \leq 1,\ i = 1, ..., m \\ & \exp({\boldsymbol{g_i}}^T\boldsymbol{y} + h_i) = 1,\ i = 1, ..., p, \end{aligned} \tag{4.12}$$

where $\boldsymbol{a}_{ik} \in \mathbb{R}^n, i = 0, ..., m$ and $\boldsymbol{g}_i \in \mathbb{R}^n, i = 1, ..., p$ contain the exponents of the posynomial inequality and monomial equality constraints, respectively. By taking the logarithm the objective and constraint functions, problem (4.12) can be equivalently transformed to the following standard convex problem

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \log\left(\sum_{k=1}^{K_0} \exp({\boldsymbol{a_{0k}}}^T\boldsymbol{y} + b_{0k})\right) \\ \text{s. t.} \quad & \log\left(\sum_{k=1}^{K_i} \exp({\boldsymbol{a_{ik}}}^T\boldsymbol{y} + b_{ik})\right) \leq 0,\ i = 1, ..., m \\ & {\boldsymbol{g_i}}^T\boldsymbol{y} + h_i = 0,\ i = 1, ..., p. \end{aligned} \tag{4.13}$$

It is noted that the log-sum-exp function: $\log\left(\sum_{k=1}^{K_i} \exp({\boldsymbol{a_{ik}}}^T\boldsymbol{y} + b_{ik})\right)$ is convex on $\mathbb{R}^n$.

4.1.4 Successive Convex Approximation

Although geometric program can be transformed into a standard convex problem, many non-convex problems cannot be solved by using this technique. One general approach to tackle a non-convex optimization problem is to approximate it with a convex optimization problem and solve the approximated problem iteratively where the objective and/or constraint functions are approximated by corresponding convex/concave functions. Specifically, these approximated convex problems are solved iteratively, and the solution obtained in each iteration is used to obtain the new, usually better, approximations of the objective and constraint functions in the next iteration.

An important property of SCA approach is that it guarantees the solutions of the series of approximations converge to a point satisfying the KarushKuhn-Tucker (KKT) conditions of the original problem [90]. In particular, for a general non-convex problem without equality constraints whose objective function $f_0(\boldsymbol{x})$ is convex and inequality constraint functions $f_i(\boldsymbol{x}),\ i > 0$ are non-convex, let $\tilde{f}_i(\boldsymbol{x})$ denote the approximated function of $f_i(\boldsymbol{x})$. Then, the approximated functions in SCA approach must satisfy the following conditions [90]:

$$\begin{aligned} &\textit{Condition 1}: \quad f_i(\boldsymbol{x}) \leq \tilde{f}_i(\boldsymbol{x}),\ \forall \boldsymbol{x}, \\ &\textit{Condition 2}: \quad f_i(\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{x}^{(q)}) = \tilde{f}_i(\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{x}^{(q)}), \\ &\textit{Condition 3}: \quad \nabla f_i(\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{x}^{(q)}) = \nabla \tilde{f}_i(\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{x}^{(q)}). \end{aligned} \tag{4.14}$$

where $\boldsymbol{x}^{(q)}$ is the optimal solution obtained by solving the approximated problem at iteration q.

4.1.5 Difference of Convex Functions (DC) Programming

DC programming can be used to solve a particular family of non-convex problems [91]. In the wireless domain, the transmission rate in many practical scenarios has the DC form which is a feature of DC programming problem (DCP). There are several popular techniques such as branch-and-bound and cutting planes algorithms to solve optimization problems, but in general, they are inefficient [92]. One desirable aspect of DCP that it is generally possible to build the approximated function satisfying three conditions in (4.14) and thus one can obtain the local optimal solutions [93]. Further details about the DC approach are briefly presented as follows.

DC function: A function $f : \mathbb{R}^n \to \mathbb{R}$ is a DC function if there exist convex functions $g, h : \mathbb{R}^n \to \mathbb{R}$ such that f can be expressed as the difference between g and h as

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) - h(\boldsymbol{x}). \tag{4.15}$$

DCP: A problem is a DCP if it has the following form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s. t.} \quad & f_i(\boldsymbol{x}) \leq 0,\ i = 1, ..., m, \end{aligned} \tag{4.16}$$

where the function $f_i : \mathbb{R}^n \to \mathbb{R}$ is a differentiable DC function for $i = 0, 1, ..., m$.

Approximated DCP: Using the first Taylor approximation, the convex term $h_i(\boldsymbol{x})$ in $f_i(\boldsymbol{x})$ can be approximated by its lower-bound as follows

$$h_i(\boldsymbol{x}) \geq \tilde{h}_i(\boldsymbol{x}, \boldsymbol{x}^{(q)}) = h_i(\boldsymbol{x}^{(q)}) + \nabla h_i^T(\boldsymbol{x}) \left(\boldsymbol{x} - \boldsymbol{x}^{(q)}\right). \tag{4.17}$$

Then, we have $f_i(\boldsymbol{x}) = g_i(\boldsymbol{x}) - h_i(\boldsymbol{x}) \leq g_i(\boldsymbol{x}) - \tilde{h}_i\left(\boldsymbol{x}, \boldsymbol{x}^{(q)}\right) = \tilde{f}_i\left(\boldsymbol{x}, \boldsymbol{x}^{(q)}\right)$. Using these upper-bound functions of DC functions, the approximated DCP at iteration $q + 1$ which is a standard convex problem, can be expressed as

$$\begin{aligned} \min_{\boldsymbol{x},\eta} \quad & \eta \\ \text{s. t.} \quad & \tilde{f}_0\left(\boldsymbol{x}, \boldsymbol{x}^{(q)}\right) - \eta \leq 0, \\ & \tilde{f}_i\left(\boldsymbol{x}, \boldsymbol{x}^{(q)}\right) \leq 0,\ i = 1, ..., m. \end{aligned} \tag{4.18}$$

4.2 Massive MIMO Technology

The MIMO wireless technology bring many benefits such as inter-user interference mitigation and capacity enhancement and this technology has been studied actively over the years. It plays important role in enabling the MEC system and its variant. In this section, we describe some fundamentals of the MIMO technology, in particular the lower capacity bound for zero-forcing (ZF) detection in two cases: perfect and imperfect channel state information (CSI), as presented in [31]. Let us consider the uplink of a multi-user (MU)-MIMO system that includes one BS equipped with an array of M antennas receiving data from K single-antenna user equipments (UEs). Let $\boldsymbol{h}_k \in \mathbb{C}^{M \times 1}$ denote the uplink channel vector between UE k and the BS's antennas where elements of the uplink channel vector $\boldsymbol{h}_k$ are modeled as $h_{mk} = \varphi_{mk}\sqrt{\beta_k}, m \in \{1, 2, ..., M\}$, where φ_{mk} and β_k represent the small-scale and large-scale fading coefficients, respectively. Assuming that the channel is slow fading and the small-scale channel coefficients $\varphi_{mk}, \forall m$ are independently and identically distributed (i.i.d.) $\mathcal{CN}(0, 1)$ random variables. Let p_k denote the uplink transmit power of UE k and $\boldsymbol{n}$ denote the noise vector whose components are i.i.d. $\mathcal{CN}(0, \sigma_{bs})$ variables.

4.2.1 Perfect CSI (P-CSI) Estimation

The received baseband signal at the BS after passing through a linear detector is

$$y_k = \sqrt{p_k}\boldsymbol{a}_k^H \boldsymbol{h}_k x_k + \sum_{i \neq k, i \in \mathcal{K}} \sqrt{p_i}\boldsymbol{a}_k^H \boldsymbol{h}_i x_i + \boldsymbol{a}_k^H \boldsymbol{n}, \tag{4.19}$$

where x_k represents the transmitted symbol from UE k, which satisfies $\mathbb{E}(|x_k|^2) = 1$, and $\boldsymbol{a}_k$ denotes the receive combining vector of UE k, and $\mathcal{K}_\xi$ is the set of UEs. The uplink average rate achieved by UE k is given by

$$r_k = \mathbb{E}\left(W \log_2(1 + \gamma_k)\right), \tag{4.20}$$

where W is the communication bandwidth and γ_k is the signal-to-noise-plus-interference ratio (SINR) of UE k, which can be written as

$$\gamma_k = \frac{p_k |\boldsymbol{a}_k^H \boldsymbol{h}_k|^2}{\sum\limits_{i \neq k, i \in \mathcal{K}_\xi} p_i |\boldsymbol{a}_k^H \boldsymbol{h}_i|^2 + \sigma_{bs} ||\boldsymbol{a}_k||^2}. \tag{4.21}$$

The linear detector matrix corresponding to the ZF method can be written as $\boldsymbol{A} = \boldsymbol{H}(\boldsymbol{H}^H \boldsymbol{H})^{-1}$ where $\boldsymbol{H}$ is a channel matrix whose columns are channel vectors of the UEs and $\boldsymbol{A}$ is the combining matrix whose columns are combining vectors of the UEs. Then, the lower bound on the achievable rate with P-CSI is given by [25]

$$r_k^{\mathsf{lb}} = W \log_2 \left(1 + \frac{p_k/\sigma_{bs}}{\mathbb{E}\left(\left[(\boldsymbol{H}^H \boldsymbol{H})^{-1}\right]_{kk}\right)}\right) = W \log_2(1 + p_k \beta_k^{\mathsf{a}}), \tag{4.22}$$

where $\beta_k^{\mathsf{a}} = \frac{(M - |\mathcal{K}_\xi|)\beta_k}{\sigma_{bs}}$.

4.2.2 Imperfect CSI (IP-CSI) Estimation

Assuming that each UE employs a pilot signal to estimate its CSI once in each channel coherence interval. Let T denote the number of symbol periods corresponding to the channel coherence interval and let τ denote the number of symbols in the pilot. Let $\sqrt{\tau p^{\mathsf{tr}}}\boldsymbol{\phi}_k \in \mathbb{C}^{\tau \times 1}$ be the pilot sequence assigned for UE k where p^{tr} denotes the pilot power and $||\boldsymbol{\phi}_k||^2 = 1$. Assuming that $\tau \geq |\mathcal{K}|$ and

the pilot sequences are designed pair-wise orthogonally, i.e., $\boldsymbol{\phi}_k^H \boldsymbol{\phi}_j = 0, \forall k \neq j$. Suppose that we employ the minimum mean square error (MMSE) CSI estimation approach [25], the variance of the estimated CSI $\hat{h}_{m,k}$ is $\mathbb{E}(|\hat{h}_{mk}|^2) = \frac{\tau p^{\mathsf{tr}} \beta_k^2}{\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}}$.

Let $\boldsymbol{\epsilon} = \hat{\boldsymbol{H}} - \boldsymbol{H}$ be the difference between the estimated channel matrix $\hat{\boldsymbol{H}}$ and true channel matrix $\boldsymbol{H}$. Following [25], its element can be modeled as $\epsilon_{ik} \sim \mathcal{CN}(0, \frac{\sigma_{bs}\beta_k}{\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}})$. The received signal associated with the UE k after passing through the ZF-based detector $\hat{\boldsymbol{A}} = \hat{\boldsymbol{H}} (\hat{\boldsymbol{H}}^H \hat{\boldsymbol{H}})^{-1}$ in this scenario can be re-written as

$$y_k = \sqrt{p_k} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{h}}_k x_k + \sum\nolimits_{i \neq k, i \in \mathcal{K}_\xi} \sqrt{p_i} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{h}}_i x_i - \sum\nolimits_{i \in \mathcal{K}_\xi} \sqrt{p_i} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{\epsilon}}_i x_i + \hat{\boldsymbol{a}}_k^H \boldsymbol{n}. \tag{4.23}$$

By treating the estimated channel as the true channel, the average rate achieved by UE k can be computed as $\hat{r}_k = \mathbb{E}(W \log_2(1 + \hat{\gamma}_k))$, where $\hat{\gamma}_k$ is expressed as [25]

$$\hat{\gamma}_k = \frac{p_k |\hat{\boldsymbol{h}}_k^H \hat{\boldsymbol{a}}_k|^2}{\sum\limits_{i \neq k, i \in \mathcal{K}_\xi} p_i |\hat{\boldsymbol{h}}_i^H \hat{\boldsymbol{a}}_k|^2 + |\hat{\boldsymbol{a}}_k|^2 \sum\limits_{i \in \mathcal{K}_\xi} \frac{p_i \sigma_{bs} \beta_i}{\tau p^{\mathsf{tr}} \beta_i + \sigma_{bs}} + \sigma_{bs} ||\hat{\boldsymbol{a}}_k||^2}. \tag{4.24}$$

Then, with ZF combining based detection, the lower bound of average rate $\hat{r}_k^{\mathsf{lb}}$ using the Jensen's inequality can be written as follows [25]:

$$\begin{aligned} \hat{r}_k^{\mathsf{lb}} &= W \log_2 \left(1 + \frac{1}{\mathbb{E}(\hat{\gamma}_k^{-1})}\right) = W \log_2 \left(1 + \frac{p_k}{\sum_{i \in \mathcal{K}} p_i \lambda_{k,i} + \sigma_k}\right) \\ &= W \log_2 \Big(\boldsymbol{p}^{\mathsf{t}} \boldsymbol{\lambda}_k + \sigma_k + p_k\Big) - W \log_2 \Big(\boldsymbol{p}^{\mathsf{t}} \boldsymbol{\lambda}_k + \sigma_k\Big), \end{aligned} \tag{4.25}$$

where $\lambda_{k,i} = \frac{(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}) \sigma_{bs} \beta_i}{(\tau p^{\mathsf{tr}} \beta_i + \sigma_{bs}) \tau p^{\mathsf{tr}} \beta_k^2 (M - K_\xi)}$, $\sigma_k = \frac{(\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}) \sigma_{bs}}{\tau p^{\mathsf{tr}} \beta_k^2 (M - K_\xi)}$.

4.3 Computation Task Models

In the MEC system, computation resources of different capacity are available at mobile devices, cloudlets, and cloud. Therefore, efficient distribution and processing of the computation workloads from different wireless applications play plays a vital role in designing MEC systems. A general model to describe computation tasks or workload is a non-trivial task when one wants to achieve the broad applicability over different practical applications and mathematical tractability. Various

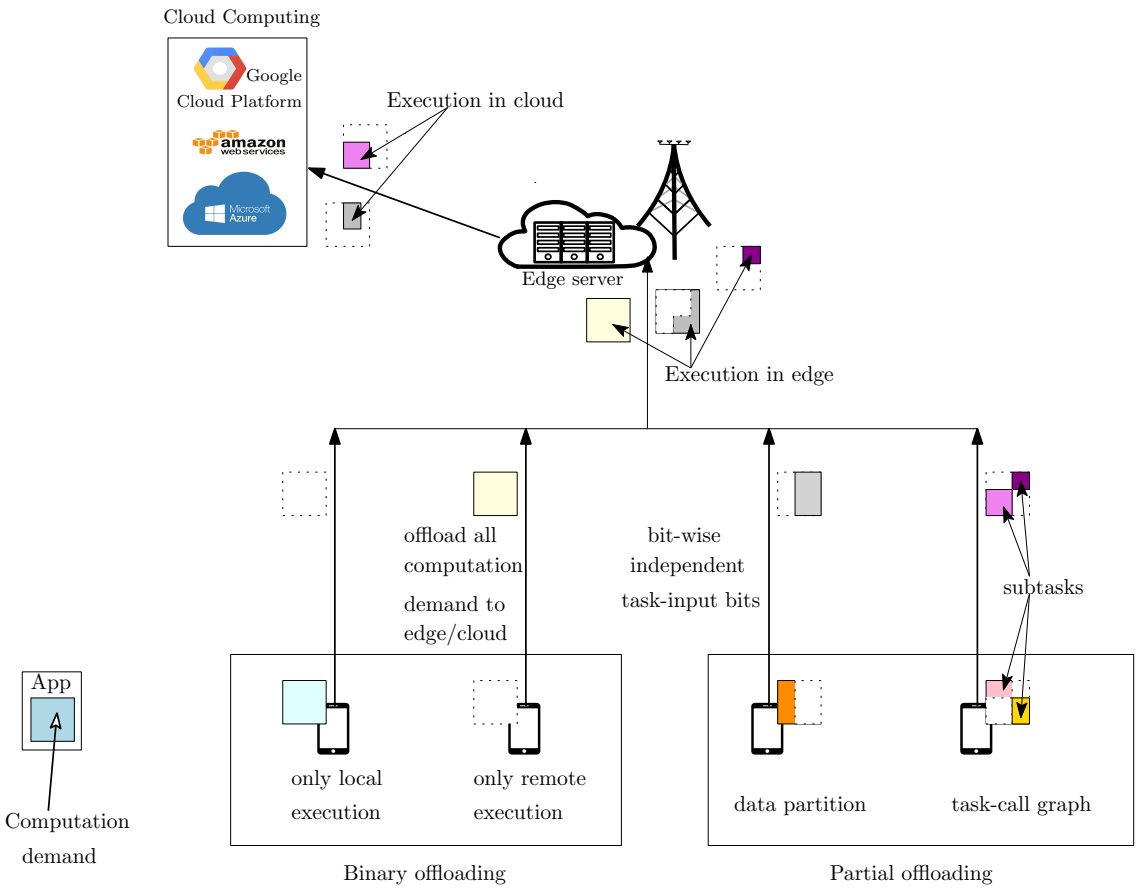

Figure 4.1 – Computation task models.

factors such as context awareness and generality, simplicity, and tractability must be considered in such the model, which must capture the essentials of an application and offer meaningful insights for engineering practice, from the MEC design perspective. In particular, the computation workload of a specific application can be partitioned into sub-tasks in certain practical scenarios and applications [94–97]. Accordingly, the two offloading mechanisms which are binary and partial offloading as shown in Fig. 4.1, have been used in the management of the computation workloads and the offloading decision making process. This section briefly introduces the task models for binary and partial offloading.

4.3.1 Task Model for Binary Offloading

The binary offloading model must be used for computation demand from a compact task that cannot be partitioned and must be executed entirely either in the local end-device or in a remote cloud or edge server [52, 98]. Such a task can be parameterized by the task input-data size (in

bits), the computation workload (CPU cycles), the completion deadline (seconds), and the task output-data size (in bits). These parameters are related to the nature of the applications and can be estimated through task profilers [22, 95, 99]. In the existing literature, the relation between the computation workload and the task input-data size is captured by using the probabilistic and deterministic relational models. In particular, the number of CPU cycles needed to execute 1-bit of task input data is modeled as a random variable in case of probabilistic case [98], and as a fixed relation in case of deterministic case [100].

4.3.2 Task Models for Partial Offloading

In practice, many mobile applications have computation demand captured by multiple procedures/tasks. For example, the action recognition application for videos can be decomposed into two main tasks, the first one for capturing the spacial information and the second one for analyzing the temporal information [101]. The partial offloading can be applied in this scenario where a part of each arrival computation demand is locally processed at end devices and the remaining part is remotely executed at cloud/cloudlet servers.

One simple task model for partial offloading is the data-partition model, where the task-input bits are bit-wise independent and they can be divided into different sizes and executed by different entities in MEC systems, e.g., parallel execution at the mobiles and MEC server [4]. In this model, the computation workload is proportional to the data size [4]. Another task model for partial offloading is the task-call graph, which can capture the number of CPU cycles and inter-dependency among different procedures/subtasks in an application [102]. For the task-call graph model, the computation workload of a particular application is captured by subtasks where each subtask is also parameterized by the task input-data size (in bits), the computation workload (CPU cycles), the completion deadline (seconds), and the task output-data size (in bits).

4.4 5G Resource Blocks

The general 5G NR frame struture can be shown in Fig. 4.2. In the time domain, NR transmissions are organized into frames of length 10 ms. Each frame is divided into 10 subframes of length 1 ms. A subframe is divided into slots consisting of 14 OFDM symbols, and the duration of a slot depends

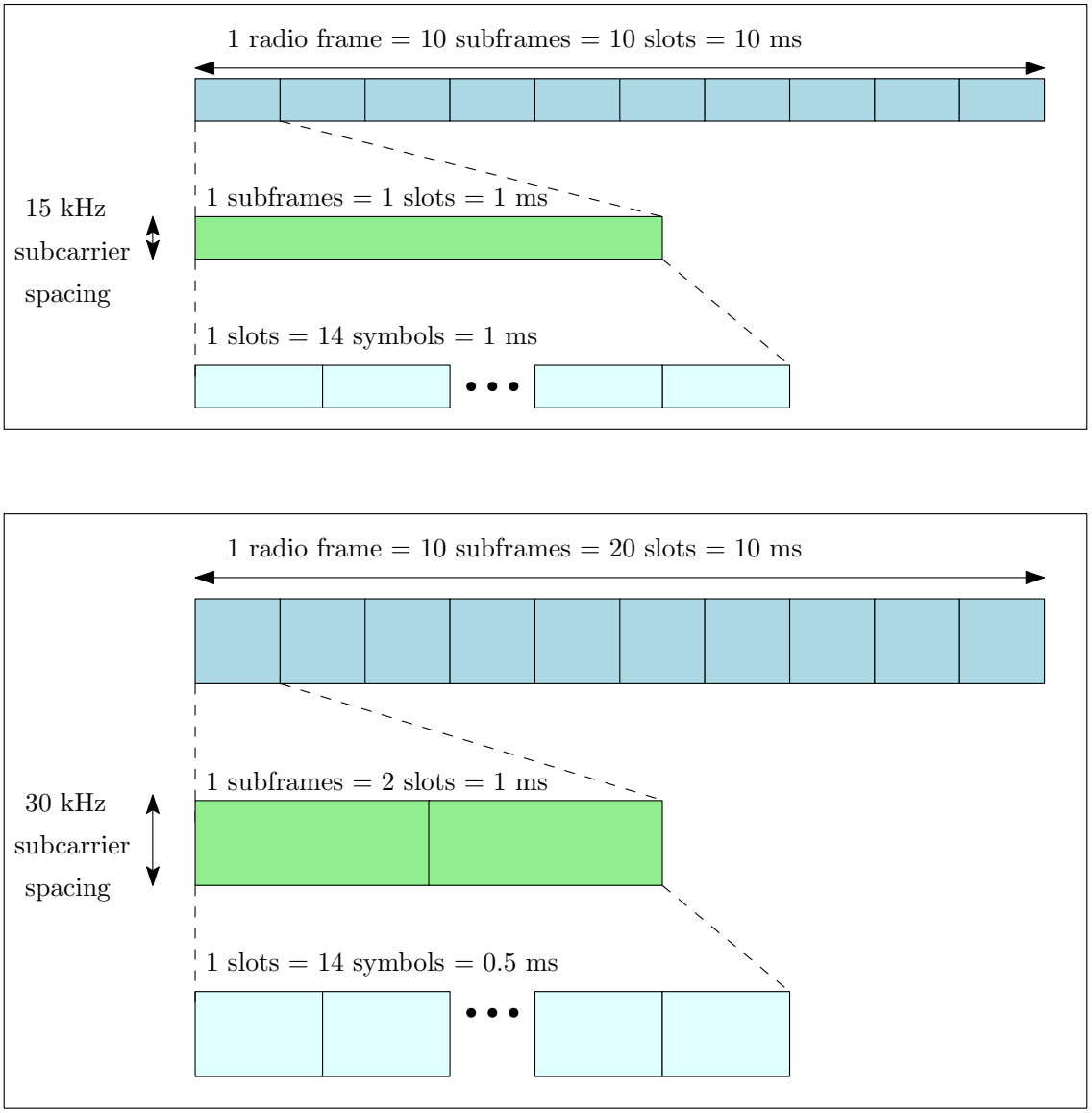

Figure 4.2 – 5G NR frame structure.

on the numerology [24]. A slot can be divived into mini-slots of 2, 4, or 7 OFDM symbols [23, 103].

Resource element (RE) is defined as one sub-carrier over one OFDM symbol [104]. Resource element is the smallest time-frequency resource unit used for downlink/uplink transmission in NR. Furthermore, 12 consecutive subcarriers in the frequency domain are called a resource block. The time duration of resource block is flexible and depends on the numerology. Resource allocation to support transmissions of different wireless users should be performed in units of resource blocks [4]. For slot based scheduling, the resource block in numerology μ has the subcarrier bandwidth of $2^{\mu} \times 15$ kHz and the time duaration of a slot. For non-slot based scheduling, the resource block in

numerology μ has the subcarrier bandwidth of $2^{\mu} \times 15$ kHz and the time duaration of a mini-slot [105].

By allowing flexible numerology, 5G NR can provide different services with different QoS requirement. For example, lower numerologies are more suitable for mMTC, since they can support higher number of simultaneously connected devices within the same bandwidth and require lower power, intermediate numerologies are appropriate for eMBB which requires both, high data rate and significant bandwidth, and higher numerologies are more suitable for delay-sensitive applications due to shorter symbol duration [106]. In particular, for slot-based scheduling, the transmission duration of 1 ms, 0.5 ms, 0.25 ms, 0.125 ms, and 0.0625 ms is achieved when subcarrier spacing are 15 kHz, 30 kHz, 60 kHz, 120 kHz, and 240 kHz, respectively. Moreover, by allowing larger subcarrier spacing, 5G NR allows the channel bandwidth up to 100 MHz, and the modulation processing complexity does not increase exponentially for wider bandwidths [107]. It is noted that the maximum channel bandwidth in LTE is 20 MHz.

Chapter 5

Computation Offloading in MIMO Based MEC Systems Under Perfect and Imperfect CSI Estimation

The content of this chapter was published in IEEE Transactions on Services Computing in the following paper:

Ti Ti Nguyen, Long B. Le, and Quan Le-Trung "Computation Offloading in MIMO Based Mobile Edge Computing Systems Under Perfect and Imperfect CSI Estimation," *IEEE Trans. Serv. Comput.,* Early Access, Jan. 2019.

5.1 Abstract

Intelligent offloading of computation-intensive tasks to a mobile cloud server provides an effective mean to expand the usability of wireless devices and prolong their battery life, especially for low-cost IoT devices. However, realization of this technology in MIMO systems requires sophisticated design of joint computation offloading and other network functions such as CSI estimation, combining, and resource allocation. In this paper, we study the computation task offloading and resource allocation optimization in MIMO based MEC systems considering perfect/imperfect-CSI estimation. Our

design aims to minimize the maximum weighted energy consumption subject to practical constraints on available computing and radio resources and allowable latency. The optimal and low-complexity algorithms are proposed to solve the underlying MINLP problems. For the perfect-CSI, we employ bisection search to find the optimal solution. The low-complexity algorithms are developed by decomposing the original optimization problem into the offloading optimization (OP) and power allocation (PA) subproblems and solve them iteratively. Moreover, the difference of convex functions (DC) method is employed to deal with non-convex structure of (PA) in the imperfect-CSI scenario. Numerical results confirm the advantages of proposed designs over conventional local computation strategies in energy saving and fairness.

5.2 Introduction

The number of novel and sophisticated wireless applications has increased drastically in recent years such as object recognition, social network, e-health, natural language processing, and virtual reality gaming thanks to the appearance of artificial intelligence-based services and high-speed communications [108]. It is expected that these desktop-level applications can be run on mobile platforms equipped with powerful processors [109]. However, this is challenging due to limitation of energy and computation capacity on mobile devices, especially for low-cost IoT devices. In fact, deployment of higher clock frequency central processing units (CPU) to process computation-intensive applications leads to the increase in the energy consumption [26]. Unfortunately, the advancement in mobile battery technology can improve the energy density only sixfold since the 1900s [110] while the computing capacity of the mobile chipset increases exponentially following Moorse's law. This means that the improvement in the battery capacity is not sufficiently fast to keep up with the practical applications' requirements.

Thus, the battery can become the bottleneck to realize many emerging mobile services. One potential solution for enhancing the mobile usability and extending the mobile battery life is to offload the computation-intensive tasks from mobile users to the central cloud or edge-cloud using the so-called mobile cloud computing (MCC) or mobile edge computing (MEC) technologies [111]. The computation offloading in both paradigms is quite similar and the difference between them is related to the available computing resources and the relative distance from the cloud to mobile users. For the MCC, the enormous computing resource is provided in the core network, whereas the MEC

system is equipped with more limited computing resource at the network edges. Moreover, the MCC architecture usually has limited backhaul capacity, thus it may not be suitable for applications with strict latency constraints. The MEC can address this challenge but the limited computing resource at the edge-cloud must be carefully allocated to efficiently support the computation offloading services.

5.2.1 Related Works

Literature survey of recent computation offloading designs can be found in [22]. A mobile cloud middleware is proposed in [112] to enable the interoperability across multiple clouds, asynchronous delegation of mobile tasks, and dynamic allocation of the cloud resource. Moreover, the MAUI platform in [95] enables fine-grained code offload by automatically extracting the program state needed in the offloading process and creating the replicated version of the user's application execution file in the cloud. The offloading decision optimization is recently studied in [113, 114]. In these existing works, the offloading decision is not jointly optimized with constrained wireless resource allocation. However, this is very important for the setting where multiple users request services at the same time, which may create strong interference if not be well managed and low transmission rates, which may lead to failure of the offloading process.

In order to guarantee the service latency requirements and successful transmission of the involved data to the cloud in an energy-efficient manner, it is important to jointly optimize the allocation of computation and radio resources [52]. Along this line, Zhang et al. in [28] consider probabilistic computation offloading and study an optimal strategy to minimize the energy consumption considering data transmission over a Gilbert-Elliott channel. In [44], the authors formulate the link selection and data transmission scheduling as a discrete-time stochastic dynamic program to optimize the energy efficiency for the single-task single-user system. Computation offloading for the multi-task single-user system is considered in [42], [43]. In particular, the task offloading is dynamically decided by using the Lyapunov optimization method in [42] or by solving a constrained shortest path problem in [43].

For the single-task multi-user system, the computation offloading and resource allocation optimization becomes more complicated [15, 34, 45–47]. Partial computation offloading to minimize the weighted sum of energy consumption is studied in [34], where CPU partitioning decision variables

are assumed to be continuous. In [15, 45, 48], the authors consider the single-task offloading and resource allocation for the interference-free transmission scenario. In [45], the computing and communication resources are shared among users with the control assistance of the network operator. This proposed framework exploits available network resources with small communication delay to achieve high energy-efficiency. In [15], the authors consider the mixed fog/cloud architecture to leverage the computing resources both at the fog and the cloud while the work [48] employs the Lyapunov-based optimization technique to minimize the average weighted sum power consumption where the computation tasks of mobile users are assumed to be fine-grained.

Moreover, iterative algorithms are developed in [46, 47] to improve energy-efficiency considering radio interference. Specifically, the decomposition framework and heuristic algorithm are respectively proposed in [46] and [47] to deal with the complicated intra- and inter-cell interference. However, making offloading decisions based on the priority in [47] may not guarantee fairness among mobile users. Some recent works consider the integration of the massive MIMO technology into the MCC/MEC system [35, 115, 116]. In particular, they tackle the joint backhaul, computing and radio resource allocation problem for a single-task offloading problem which aims to minimize the total energy consumption. These designs, however, do not consider the CSI estimation performance, which is a very important factor directly affecting the wireless communication quality and transmission rate of the underlying MIMO based wireless system [117].

While the above papers consider three different MEC systems, namely the single-task single-user system, single-task multi-user system, and multi-task single-user system, the more complex multi-task multi-user system has been studied in some recent works [49–51]. Specifically, [49] proposes a heuristic algorithm to tackle the offloading and task scheduling problems for multicore-based mobile devices where data transmission is not optimized. Assuming the interference-free wireless network, binary task offloading design is conducted in [50] by employing semi-definite relaxation and the probability based rounding technique. However, both transmit energy and spectral efficiency are assumed to be unchanged in optimization of the allocated bandwidth in this work, which may not hold true in practice. This optimization framework is further extended in [51] to consider co-channel interference in the heterogeneous network based MEC system; however, independent power allocation for different users adopted by this work may not be efficient in managing the interference. Even though there have been some efforts in tackling the computation offloading design

for the multi-task multi-user MEC system, consideration of advanced communication aspects such as MIMO communications for such design requires much more further research.

5.2.2 Contributions and Organization of the Paper

Existing computation offloading designs for the multi-task multi-user scenario have not considered the important MIMO communication technology and its related issues such as the imperfect CSI estimation. Our current paper aims to fill this gap in the literature by proposing general offloading and resource allocation algorithms which can provide fairness and consider the cutting-edge MIMO technology. In particular, the main contributions of this paper can be summarized as follows:

- We formulate the joint computation offloading and resource allocation problem that minimizes the maximum weighted consumed energy (Min-max W.C.E) for mobile users considering the latency and resource limitation constraints. The problem formulation captures the general partial offloading for the multi-task multi-user setting where the computation tasks can be either processed locally at the mobile user or offloaded and processed in the cloud. We consider two important scenarios with perfect-CSI (P-CSI) and imperfect-CSI (IP-CSI) estimation for the MIMO-based MEC system. To the best of our knowledge, the study of P-CSI and IP-CSI for MIMO communications in the MEC system has not been conducted in the literature.

- We propose different efficient algorithms to solve the underlying MINLP problems. For the P-CSI scenario, we propose an optimal algorithm achieving the global optimal solution by employing the bisection search method in which the optimization problem is decomposed into independent convex subproblems for individual users. We also propose a low-complexity algorithm which iteratively solves the offloading subproblem (OP) and power allocation subproblem (PA) until convergence. For the IP-CSI setting, the decomposition of the original problem into the (OP) and (PA) is also performed and the DC optimization approach is then employed to convexify and tackle the non-convex constraints in the (PA) subproblem.

- We prove the convergence of different proposed iterative algorithms. Moreover, we discuss the extension of the proposed design to consider both uplink data transmission and downlink feedback of the computation outcome in the computation offloading design. We show how

our proposed algorithm can be extended to address this more general problem. Moreover, we analyze the complexity of the proposed algorithms.

- Numerical studies show that the low-complexity algorithm achieves close-to-optimal performance in the P-CSI scenario. Moreover, we show that our proposed design can achieve good fairness for different users. Finally, we investigate the impacts of different parameters including the maximum allowable delay, the energy coefficient of mobile devices, the number of computation tasks to the achievable performance.

The remaining of this paper is organized as follows. Section 5.3 presents the system model and the task scheduling and computation-resource allocation problems. Section 5.4 describes the joint radio resource and computing resource algorithms for the P-CSI scenario. Section 5.5 discusses the algorithm design for the IP-CSI scenario. Section 5.6 presents the extension of our design for applications that require downlink transmission of the computation results and analyzes the complexity of the proposed algorithms. Section 5.7 evaluates the performance of the proposed algorithms and Section 5.8 concludes the work. In Table 1, we summarize important notations in this paper.

5.3 System model and problem formulation

We consider an MEC system comprising K single-antenna users or user equipments (UEs) and one base station (BS) equipped with M antennas. For convenience, we denote the set of UEs as $\mathcal{K} = \{1, 2, ..., K\}$. We assume that $M > K$, the cloud server is located at the edge of the cellular network and the high-speed fiber link is used to connect the network operator and the cloud server. Then, the cloud can serve offloaded computation demands from multiple UEs simultaneously. It is assumed that the cloud has received in advance the UEs' task images (i.e., the replicated versions of the execution files corresponding to the offloading tasks), which can, therefore, be executed in the cloud in the offloading case [26, 46].

5.3.1 Computation Offloading Model

We assume that UE k has the set of $\mathcal{L}_k$ independent computation tasks from his/her application and these tasks can be executed locally at the mobile device or offloaded and executed in the cloud independently with the maximum allowable delay η_k [22], [102]. For example, the action recognition application for videos can be decomposed into two main tasks, the first one for capturing the spacial information and the second one for analyzing the temporal information [101]. Moreover, each task $l_k \in \mathcal{L}_k$ needs a number of CPU cycles c_{k,l_k} and a number of data bits b_{k,l_k} (to transmit the programming states to the BS). For instance, according to [95] the number of bits and CPU cycles of the task of detecting and extracting faces are about 0.26 Gcycles and 14 kbits, respectively.

We now introduce a binary offloading decision variable s_{k,l_k} for task $l_k \in \mathcal{L}_k$ as follows:

$$s_{k,l_k} = \begin{cases} 1, & \text{if task } l_k \text{ is executed at the mobile device.} \\ 0, & \text{if task } l_k \text{ is offloaded to the cloud.} \end{cases} \tag{5.1}$$

The local computation energy and time due to user k can be expressed, respectively, as

$$\xi_k^{\mathsf{lo}} = \alpha_k f_k^2 \sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}, \tag{5.2}$$

$$t_k^{\mathsf{lo}} = f_k^{-1} \sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}, \tag{5.3}$$

where α_k denotes the energy coefficient specified in the CPU model and f_k denotes the CPU clock speed of UE k (CPU cycles/second or Hz), which is assumed to be smaller than the UE's maximum clock speed F_k. Each individual UE k can partially or totally offload its computation tasks to the cloud if the underlying program can be more efficiently executed in the cloud or requires more computing resource at the mobile device to execute within the delay η_k.

5.3.2 Cloud Computation Model

Upon receiving the offloading demand from UE k, the cloud server will assign the computing resource measured in the CPU clock speed f_k^{c} to execute the UE's application. The required execution time

Table 5.1 – Important Notations

Notations	Description
$K/\mathcal{K}$	Number/set of users
$\mathcal{K}_\xi$	Set of users having at least 1 offloaded task
$\mathcal{L}_k$	Set of tasks of user k
W	Transmission bandwidth (Hz)
M	Number of antennas at BS
$\boldsymbol{H}/\hat{\boldsymbol{H}}$	Perfect/imperfect estimated channel matrix
$\boldsymbol{A}/\hat{\boldsymbol{A}}$	combining matrix with P/IP- CSI
$c_{k,l_k}/b_{k,l_k}$	Number of CPU cycles/upload bits of task l_k of UE k
b^{dl}_{k,l_k}	Number of download bits of task l_k of UE k (bits)
Γ_{dpu}	Ratio of download bits per upload bits
s_{k,l_k}	Offloading decision of task l_k of UE k
f^{c}_k	Allocated CPU clock speed for UE k from cloud (Hz)
F^{c}	Maximum CPU clock speed of cloud (Hz)
f_k/F_k	CPU clock speed of UE k and its maximum (Hz)
$p_k/p_{k,0}$	Uplink transmit power/circuit power of UE k (Watts)
p^{dl}_k	Downlink transmit power allocated for UE k (Watts)
$p_k/p^{\mathsf{dl}}_{\mathsf{max}}$	Maximum transmit power of UE k/BS (Watts)
ξ	Min-max weighted consumed energy
$\xi^{\mathsf{lo}}_k/\xi^{\mathsf{t}}_k$	Computation/transmit energy of UE k (Joule)
$t^{\mathsf{lo}}_k/t^{\mathsf{t}}_k$	Computation/transmit time of UE k (second)
α_k	Computation energy coefficient of UE k
β_k	Large-scale fading coefficient of channel $\boldsymbol{h}_k$
β^{a}_k	Defined in (5.13)
η_k	Maximum latency requirement of UE k (second)
σ_{bs}	Noise power received at BS (Watts)
σ^{dl}_k	Noise power received at UE k (Watts)
T/τ	Channel coherence interval / pilot sequence length
$r^{\mathsf{lb}}_k/\hat{r}^{\mathsf{lb}}_k$	Lower-bound of uplink rate with P/IP-CSI (bits/s)
$\hat{r}^{\mathsf{dl,lb}}_k$	Lower-bound of downlink rate (bits/s)
$\boldsymbol{\lambda}_k/\sigma_k$	Coefficients defined in (16)
$b^{\mathsf{a}}_k/c^{\mathsf{a}}_k$	Offloaded bits/ CPU cycles of UE k
ξ^{a}_k	Relation of transmission bits, energy defined in $(\mathcal{P}_3)'_k$
$\mathbb{E}(x)$	Expected value of x

to finish the computation demand from UE k in the cloud server can be expressed as

$$t^{\mathsf{c}}_k = (f^{\mathsf{c}}_k)^{-1} \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) c_{k,l_k}. \tag{5.4}$$

In addition, we assume that the total computing resources allocated by the cloud server must be within its available computing budget F^{c}. This constraint can be expressed as $\sum_{k\in\mathcal{K}_\xi} f^{\mathsf{c}}_k \leq F^{\mathsf{c}}$,

where $\mathcal{K}_\xi$=$\{k \in \mathcal{K} | \sum_{l \in \mathcal{L}_k} s_{k,l_k} < |\mathcal{L}_k|\}$, denotes the set of UEs that cannot execute all their tasks locally.

5.3.3 Wireless Transmission Model

Let $\boldsymbol{h}_k \in \mathbb{C}^{M \times 1}$ denote the uplink channel vector between UE k and the BS's antennas where elements of the uplink channel vector $\boldsymbol{h}_k$ are modeled as $h_{mk} = \varphi_{mk}\sqrt{\beta_k}, m \in \{1, 2, ..., M\}$, where φ_{mk} and β_k represent the small-scale and large-scale fading coefficients, respectively. Assuming that the large-scale channel coefficient does not change during the required task execution latency (i.e., slow fading) and the small-scale channel coefficients $\varphi_{mk}, \forall m$ are independently and identically distributed (i.i.d.) $\mathcal{CN}(0, 1)$ random variables. Let p_k denote the uplink transmit power of UE k and $\boldsymbol{n}$ denote the noise vector whose components are i.i.d. $\mathcal{CN}(0, \sigma_{bs})$ variables.

We derive the transmission rate, time, and energy for P-CSI and IP-CSI scenarios and ZF combining in the following.

5.3.3.1 P-CSI Scenario

The received baseband signal at the BS after passing through a linear detector is

$$y_k = \sqrt{p_k}\boldsymbol{a}_k^H \boldsymbol{h}_k x_k + \sum_{i \neq k, i \in \mathcal{K}_\xi} \sqrt{p_i}\boldsymbol{a}_k^H \boldsymbol{h}_i x_i + \boldsymbol{a}_k^H \boldsymbol{n}, \tag{5.5}$$

where x_k represents the transmitted symbol from UE k, which satisfies $\mathbb{E}(|x_k|^2) = 1$, and $\boldsymbol{a}_k$ denotes the combining vector of UE k.

The ergodic uplink rate achieved by UE k is given by

$$r_k = W\mathbb{E}(\log_2(1 + \gamma_k)), \tag{5.6}$$

where W is the communication bandwidth and γ_k is the signal-to-noise-plus-interference ratio (SINR) of UE k, which can be written as

$$\gamma_k = \frac{p_k |\boldsymbol{a}_k^H \boldsymbol{h}_k|^2}{\sum\limits_{i \neq k, i \in \mathcal{K}_\xi} p_i |\boldsymbol{a}_k^H \boldsymbol{h}_i|^2 + \sigma_{bs} ||\boldsymbol{a}_k||^2}. \tag{5.7}$$

Then, the average energy required for data transmission of UE k can be computed as

$$\bar{\xi}_k^{\mathsf{t}} = \mathbb{E}\left((p_k + p_{k,0})t_k\right) = (p_k + p_{k,0}) \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k} \mathbb{E}\left(\frac{1}{r_k}\right), \tag{5.8}$$

where $p_{k,0}$ denotes the circuit power. The exact expression for $\mathbb{E}(1/r_k)$ is quite complicated for analysis; therefore, we introduce the upper bound of this term as follows. Toward this end, we have following result for function $\psi(x) = 1/\log(1 + x^{-1})$:

$$\nabla^2\Big(\psi(x)\Big) = \frac{2 - (2x+1)\log\left(1 + \frac{1}{x}\right)}{\log^3\left(1 + \frac{1}{x}\right)(x^2 + x)^2} = \begin{cases} \ll 0, \text{ if } x \approx 0. \\ \approx 0, \text{ if } x \gg 0. \end{cases} \tag{5.9}$$

When UE k decides to offload its computation tasks to the cloud for energy saving, intuitively its wireless transmission condition in terms of SINR must be sufficiently good so that one can maintain the required application execution latency. Therefore, a tight approximation for the upper bound of the ergodic transmission time and energy can be obtained by applying the Jensen's inequality for a concave function as follows:

$$\mathbb{E}\left(\frac{1}{r_k}\right) \leq \left[W \log_2\left(1 + \frac{1}{\mathbb{E}\left(\gamma_k^{-1}\right)}\right)\right]^{-1} \stackrel{\text{def}}{=} \frac{1}{r_k^{\mathsf{lb}}}. \tag{5.10}$$

Then, the upper bound of the average transmission time and energy can be written, respectively, as

$$t_{k,\mathsf{P}}^{\mathsf{t,ub}} = (r_k^{\mathsf{lb}})^{-1} \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}, \tag{5.11}$$

$$\xi_{k,\mathsf{P}}^{\mathsf{t,ub}} = (p_k + p_{k,0}) t_{k,\mathsf{P}}^{\mathsf{t,ub}}. \tag{5.12}$$

Assuming that the zero-forcing (ZF) combining method, which is a popular and widely accepted technique for MIMO communication, is employed to recover the user's signal at the receiver. Then, under the P-CSI scenario, the linear detector matrix corresponding to the ZF combining method can be written as $\boldsymbol{A} = \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}$ where $\boldsymbol{H}$ is a channel matrix whose columns are channel vectors of the UEs and $\boldsymbol{A}$ is the combining matrix whose columns are combining vectors of the UEs. Then,

the lower bound on the achievable rate with P-CSI is given by [25]

$$r_k^{\mathsf{lb}} = W \log_2 \left(1 + \frac{p_k/\sigma_{bs}}{\mathbb{E}\left\{ \left[(\boldsymbol{H}^H \boldsymbol{H})^{-1} \right]_{kk} \right\}} \right) = W \log_2 (1 + p_k \beta_k^{\mathsf{a}}), \tag{5.13}$$

where $\beta_k^{\mathsf{a}} = \frac{(M - |\mathcal{K}_\xi|)\beta_k}{\sigma_{bs}}$.

5.3.3.2 IP-CSI Scenario

Assuming that each UE employs a pilot signal to estimate its CSI once in each channel coherence interval. Let T denote the number of symbol periods corresponding to the channel coherence interval and let τ denote the number of symbols in the pilot. Let $\sqrt{\tau p^{\mathsf{tr}}}\boldsymbol{\phi}_k \in \mathbb{C}^{\tau \times 1}$ be the pilot sequence assigned for UE k where p^{tr} denotes the pilot power and $||\boldsymbol{\phi}_k||^2 = 1$. Assuming that $\tau \geq |\mathcal{K}_\xi|$ and the pilot sequences are designed pair-wise orthogonally, i.e., $\boldsymbol{\phi}_k^H \boldsymbol{\phi}_j = 0, \forall k \neq j$. Suppose that we employ the minimum mean square error (MMSE) CSI estimation approach [25], the covariance of the estimated CSI $\hat{h}_{m,k}$ is $\mathbb{E}(|\hat{h}_{mk}|^2) = \frac{\tau p^{\mathsf{tr}} \beta_k^2}{\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}}$.

Let $\boldsymbol{\epsilon} = \hat{\boldsymbol{H}} - \boldsymbol{H}$ be the difference between the estimated channel matrix $\hat{\boldsymbol{H}}$ and true channel matrix $\boldsymbol{H}$. Following [25], its element can be modeled as $\epsilon_{ik} \sim \mathcal{CN}(0, \frac{\sigma_{bs}\beta_k}{\tau p^{\mathsf{tr}} \beta_k + \sigma_{bs}})$. The received signal associated with the UE k after passing through the ZF-based detector $\hat{\boldsymbol{A}} = \hat{\boldsymbol{H}} (\hat{\boldsymbol{H}}^H \hat{\boldsymbol{H}})^{-1}$ in this scenario can be re-written as

$$y_k = \sqrt{p_k} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{h}}_k x_k + \sum\nolimits_{i \neq k, i \in \mathcal{K}_\xi} \sqrt{p_i} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{h}}_i x_i - \sum\nolimits_{i \in \mathcal{K}_\xi} \sqrt{p_i} \hat{\boldsymbol{a}}_k^H \hat{\boldsymbol{\epsilon}}_i x_i + \hat{\boldsymbol{a}}_k^H \boldsymbol{n}. \tag{5.14}$$

The SINR for this IP-CSI scenario by treating the estimated channel as the true channel is expressed as [25]

$$\hat{\gamma}_k = \frac{p_k |\hat{\boldsymbol{h}}_k^H \hat{\boldsymbol{a}}_k|^2}{\sum\limits_{i \neq k, i \in \mathcal{K}_\xi} p_i |\hat{\boldsymbol{h}}_i^H \hat{\boldsymbol{a}}_k|^2 + |\hat{\boldsymbol{a}}_k|^2 \sum\limits_{i \in \mathcal{K}_\xi} \frac{p_i \sigma_{bs} \beta_i}{\tau p^{\mathsf{tr}} \beta_i + \sigma_{bs}} + \sigma_{bs} ||\hat{\boldsymbol{a}}_k||^2}. \tag{5.15}$$

Then, the ergodic rate achieved by UE k can be computed as $\hat{r}_k = \mathbb{E}(W \log_2(1 + \hat{\gamma}_k))$. With ZF based detection, the lower bound of rate $\hat{r}_k^{\mathsf{lb}}$ using the Jensen's inequality can be written as follows

[25]:

$$\begin{aligned}\hat{r}_k^{\mathsf{lb}} &= W\log_2\left(1+\frac{1}{\mathbb{E}(\hat{\gamma}_k^{-1})}\right) = W\log_2\left(1+\frac{p_k}{\sum_{i\in\mathcal{K}_\xi} p_i\lambda_{k,i}+\sigma_k}\right)\\ &= W\log_2\Big(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k+p_k\Big) - W\log_2\Big(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k\Big),\end{aligned} \tag{5.16}$$

where $\lambda_{k,i} = \frac{(\tau p^{\mathsf{tr}}\beta_k+\sigma_{bs})\sigma_{bs}\beta_i}{(\tau p^{\mathsf{tr}}\beta_i+\sigma_{bs})\tau p^{\mathsf{tr}}\beta_k^2(M-K_\xi)}$, $\sigma_k = \frac{(\tau p^{\mathsf{tr}}\beta_k+\sigma_{bs})\sigma_{bs}}{\tau p^{\mathsf{tr}}\beta_k^2(M-K_\xi)}$, and $\boldsymbol{p}^{\mathsf{t}}$ is a vector represents the transmit power of all K_ξ users.

The training is performed for each coherence bandwidth chunk B_c in the total available bandwidth of W. For each channel coherence interval of T symbol periods, UEs will spend τ symbol periods for training and the remaining $T-\tau$ symbol periods for data transmission. Then, the upper bound of the average transmission time and energy (including the training time and energy) can be written, respectively, as

$$t_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \frac{T}{T-\tau}t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{5.17}$$

$$\xi_{k,\mathsf{IP}}^{\mathsf{t,ub}} = \left(\frac{\tau(p^{\mathsf{tr}}+p_{k,0})}{T-\tau}+p_k+p_{k,0}\right)t_{k,\mathsf{IP1}}^{\mathsf{t,ub}}, \tag{5.18}$$

where $t_{k,\mathsf{IP1}}^{\mathsf{t,ub}} = (\hat{r}_k^{\mathsf{lb}})^{-1}\sum_{l_k\in\mathcal{L}_k}(1-s_{k,l_k})b_{k,l_k}$.

The total latency experienced by an offloaded task has different components, namely the time required for sending transmission bits (e.g., program states) to the cloud, the computation time in the cloud, and the time required for downloading the results to the UE from the cloud. For many practical applications, the amount of data required to report the computation outcome is typically much smaller than the amount of offloading data [46]. Therefore, we first omit the time for sending back the computation outcome from the cloud server to mobile users in our design and we will consider it in Section 5.6.

5.3.4 Problem Formulation

In this paper, we consider minimizing the energy consumption from the users' perspective to prolong their lifetime as studied in many recent works [46, 113]. Moreover, our design aims to achieve fairness for different users. Toward this end, we jointly optimize the offloading decisions, computation and radio resource allocation to minimize the maximum weighted energy consumption at mobile users considering latency and limited computation-radio resource constraints. This problem can be

formulated as follows:

$$(\mathcal{P}_1) \quad \min_{\boldsymbol{S},\boldsymbol{f},\boldsymbol{f}^{\mathsf{c}},\boldsymbol{p}} \max_{k} \quad w_k(\xi_k^{\mathsf{lo}} + \xi_k^{\mathsf{t}}) \tag{5.19a}$$

$$\text{s. t.} \quad t_k^{\mathsf{lo}} \leq \eta_k, \quad \forall k, \tag{5.19b}$$

$$t_k^{\mathsf{t}} + t_k^{\mathsf{c}} \leq \eta_k, \ \forall k, \tag{5.19c}$$

$$s_{k,l_k} \in \{0,1\}, \quad \forall k, \tag{5.19d}$$

$$\textstyle\sum_{k\in\mathcal{K}_\xi} f_k^{\mathsf{c}} \leq F^{\mathsf{c}}, \quad f_k^{\mathsf{c}} \geq 0, \tag{5.19e}$$

$$0 \leq F_k \leq F_k^{\mathsf{max}}, \quad \forall k, , \tag{5.19f}$$

$$0 \leq p_k \leq P_k^{\mathsf{max}}, \quad \forall k, \tag{5.19g}$$

where w_k denotes the energy weight of UE k, $\boldsymbol{S} = \{\boldsymbol{s}_k, \forall k\}, \boldsymbol{s}_k = \{s_{k,l_k}, \forall l_k\}$, $\{\boldsymbol{f}, \boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}\} = \{f_k, f_k^{\mathsf{c}}, p_k, \forall k\}$, η_k is the maximum allowable delay of UE k, F_k^{max} denotes the maximum computation capacity of UE k, and P_k^{max} represents the maximum transmit power of UE k, t_k^{t} and ξ_k^{t} stand for transmission time and energy, respectively, which can be expressed for the P-CSI and IP-CSI scenarios as

$$t_k^{\mathsf{t}} = \begin{cases} t_{k,\mathsf{P}}^{\mathsf{t,ub}}, \text{ P-CSI} \\ t_{k,\mathsf{IP}}^{\mathsf{t,ub}}, \text{ IP-CSI} \end{cases} \quad ; \quad \xi_k^{\mathsf{t}} = \begin{cases} \xi_{k,\mathsf{P}}^{\mathsf{t,ub}}, \text{ P-CSI} \\ \xi_{k,\mathsf{IP}}^{\mathsf{t,ub}}, \text{ IP-CSI} \end{cases} .$$

In this problem, (5.19b) captures the delay requirements for local computation while (5.19c) represents the total latency requirements for the offloaded tasks. The binary offloading decision is described in constraints (5.19d) while the limited computing resources are represented in constraints (5.19e) and (5.19f), where (5.19e) describes this constraint for the cloud and (5.19f) captures these constraints at the UEs. Finally, (5.19g) describes the UEs' maximum transmit power constraints. It is noted that we develop centralized algorithms to tackle problem $(\mathcal{P}_1)$ in this work. However, they are developed based on decomposition techniques. Thus, they can be modified to convert to decentralized algorithms.

5.3.5 Problem Transformation

To gain insights into the non-smooth min-max objective function, we introduce an auxiliary variable ξ and transform $(\mathcal{P}_1)$ to the following equivalent problem:

$$(\mathcal{P}_2) \quad \min_{\boldsymbol{S},\boldsymbol{f},\boldsymbol{f^c},\boldsymbol{p},\xi} \quad \xi \tag{5.20a}$$

$$\text{s. t.} \quad w_k(\xi_k^{\mathsf{lo}} + \xi_k^{\mathsf{t}}) \leq \xi, \ \forall k, \tag{5.20b}$$

$$(5.19b) - (5.19g).$$

We now state an important result for this transformed problem in the following proposition.

Proposition 5.1. *The optimal value of f_k is equal to $(\eta_k)^{-1} \sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k}$ if there exists a feasible set of s_{k,l_k} such as f_k is less than or equal to F_k^{max} for all UE k.*

Proof. It can be verified from (5.2) that the local energy computation of UE k increases with the CPU clock speed f_k. Therefore, this energy component achieves its minimal value at the smallest possible value of f_k. From (5.3), (5.19b) and (5.19f), one can infer that $f_k = (f_k)_{\min} = (\eta_k)^{-1} \sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k} \leq F_k^{\mathsf{max}}$ at optimality. □

From the results of *Proposition 5.1*, the local computation energy ξ_k^{lo} and constraints (5.19f) can be rewritten as the function of the offloading decision variable s_{k,l_k} as

$$\xi_k^{\mathsf{lo}} = \alpha_k \eta_k^{-2} \left(\sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k} \right)^3, \quad \forall k \in \mathcal{K}, \tag{5.21}$$

$$\sum_{l_k \in \mathcal{L}_k} s_{k,l_k} c_{k,l_k} \leq \eta_k F_k^{\mathsf{max}}, \quad \forall k \in \mathcal{K}. \tag{5.22}$$

Therefore, the objective function and constraint functions of problem $(\mathcal{P}_2)$ can be expressed in $\boldsymbol{S}, \boldsymbol{f^c}, \boldsymbol{p}, \xi$ and this problem can be recast as

$$(\mathcal{P}_2) \quad \min_{\boldsymbol{S},\boldsymbol{f^c},\boldsymbol{p},\xi} \quad \xi$$

$$\text{s. t.} \quad (5.20b), (5.19c) - (5.19e), (5.19g), (5.22).$$

This problem is difficult to solve due to the complex fractional form of the transmission time and energy, the logarithmic transmission rate function, binary variables $\boldsymbol{S}$ and continuous variables $\boldsymbol{f^c}, \boldsymbol{p}, \xi$. In the following sections, we will present our proposed algorithms to solve problem $(\mathcal{P}_2)$ for the P-CSI and IP-CSI scenarios.

5.4 Algorithm Design for P-CSI Scenario

To solve the difficult MINLP $(\mathcal{P}_2)$, we propose two algorithms where the first one (P-O) can find the global optimal solution while the second one (P-SO) achieves a solution with lower complexity. In (P-O) algorithm, we first employ the bisection search for ξ where the upper-bound $\xi_{\max}$ and lower-bound $\xi_{\min}$ of ξ are iteratively updated until the difference between them is sufficiently small. This updating mechanism is based on the feasibility verification of problem $(\mathcal{P}_2)$ for a given value of ξ as follows. If the set of constraints is feasible, then the upper-bound of the objective function will decrease and is set equal to ξ; otherwise, its lower-bound will increase and is set equal to ξ. To verify the feasibility, we decompose this problem into individual users' subproblems, then find the minimum allocated computing resource from the cloud server to each user. This is done by searching all offloading decision combinations of each user. The total required computing resource of all users and its available budget from the cloud server is compared to determine the feasibility condition.

In the (P-SO) algorithm, the original problem is decomposed into the offloading optimization (OP) and power allocation (PA) subproblems which are solved iteratively. For the (OP) subproblem, we directly find the offloading decisions via linearizing the non-convex constraints. Besides, we also propose an indirect method to tackle this problem via the search of offloading decision combinations; however, it can be done rapidly compared to the search in the (P-O) algorithm, which will be presented more detail later. For the (PA) subproblem, we apply the bisection search and in each iteration, we also compare the required and available computing resource from cloud server to verify the feasibility of convex constraints.

5.4.1 P-CSI - Optimal Algorithm (P-O)

The feasibility verification of non-convex mixed integer constraints for a given value of ξ is still very challenging. Fortunately, the lower-bound of the achievable rate in (5.13) depends only on p_k and the number of offloading UEs $|\mathcal{K}_\xi|$, which means that all constraints (5.20b), (5.19c), (5.19d), (5.19g) and (5.22) are independent for different UEs for a given value of ξ. Therefore, we can decompose this problem into the feasibility verification problems for individual UEs if we can deal with the dependent relation of UEs in constraints (5.19e). Toward this end, we first remove constraints (5.19e), determine $|\mathcal{K}_\xi|$, and then find the minimum allocated computing resource from the cloud server for each user. The constraint (5.19e) will be then verified by using the obtained computing resource allocation solution. Specifically, for a given value of ξ, UE k should upload its computation tasks if the local execution consumes the total energy greater than ξ. Therefore, the number of offloading UEs $|\mathcal{K}_\xi|$ can be computed as follows:

$$|\mathcal{K}_\xi| = \sum_k \delta_k, \text{ and } \delta_k = \begin{cases} 1, \text{ if } w_k \xi_k^{\mathsf{lo}} > \xi, \forall l_k \text{ st } s_{k,l_k}=1 \\ 0, \text{ otherwise} \end{cases}. \tag{5.24}$$

We now study the following subproblems that can be solved independently by individual UEs ($k \in \mathcal{K}_\xi$) for given values of ξ and $|\mathcal{K}_\xi|$:

$$\begin{aligned} (\mathcal{P}_3)_k \quad & \min_{\boldsymbol{s}_k, f_k^{\mathsf{c}}, p_k} \quad [f_k^{\mathsf{c}}]^+ \\ & \text{s. t.} \quad (5.20b)_k, (5.19c)_k, (5.19d)_k, (5.19g)_k, (5.22)_k, \end{aligned}$$

where $[f_k^{\mathsf{c}}]^+ = \max(f_k^{\mathsf{c}}, 0)$, constraints $(5.20b)_k$, $(5.19c)_k$, $(5.19d)_k$, $(5.19g)_k$, and $(5.22)_k$ denote the corresponding constraints (5.20b), (5.19c), (5.19d), (5.19g), and (5.22) for UE k, respectively.

Suppose that all sub-problems $(\mathcal{P}_3)_k, \forall k$ can be solved. Then, constraint (5.19e), which couples all UEs, can be addressed by using the result in the following proposition.

Proposition 5.2. *For a given value of ξ, problem $(\mathcal{P}_2)$ is feasible if all subproblems $(\mathcal{P}_3)_k, \forall k$ are feasible and $\sum_{k \in \mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}} \leq F^{\mathsf{c}}$ where $f_{k,\xi}^{\mathsf{c,min}}$ is the optimal value of $(\mathcal{P}_3)_k$.*

Proof. If all subproblems $(\mathcal{P}_3)_k, \forall k \in \mathcal{K}_\xi$ are feasible, constraints (5.20b), (5.19c), (5.19d), (5.19g), (5.22) are satisfied and the required CPU clock speeds f_k^{c} of all UEs are at their minimum; thus,

Algorithm 5.1. Optimal Algorithm - P-CSI (P-O)

1: **Initialize**: choose ϵ, $\xi_{\min} = 0$ and $\xi_{\max} = \min(\max(w_k \xi_k^{\mathsf{lo}} | s_{k,l_k} = 1, \sum_{l_k \in \mathcal{L}_k} c_{k,l_k} \leq \eta_k F_k), \xi^{\infty})$.
2: **while** $\xi_{\max} - \xi_{\min} < \epsilon$ **do**
3: Assign $\xi = (\xi_{\max} + \xi_{\min})/2$.
4: Determine set $\mathcal{K}_\xi$ as in (5.24).
5: Solve $(\mathcal{P}_3)_k$ to get $f_{k,\xi}^{\mathsf{c,min}}$ for all $k \in \mathcal{K}_\xi$.
6: Assign $feasibility = true$ if all subproblems $(\mathcal{P}_3)_k$ are feasible and $\sum_{k \in \mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}} \leq F^{\mathsf{c}}$.
7: Assign $(\xi_{\max}, \xi_{\min})$ = bisectionSearch$(feasibility, \xi)$
8: **end while**

the total CPU clock speed $\sum_{k \in \mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}}$ is also minimum. As a result, constraint (5.19e) is satisfied if the minimum required computing resources for different UEs satisfy $\sum_{k \in \mathcal{K}_\xi} f_{k,\xi}^{\mathsf{c,min}} \leq F^{\mathsf{c}}$, which then implies that problem $(\mathcal{P}_2)$ is feasible. □

Using the results in *Proposition 5.2*, we propose an optimal algorithm to solve problem $(\mathcal{P}_2)$ as described in Algorithm 5.1. In this algorithm, the offloading decisions and the allocation of cloud computing resource can be decided by the UEs. Moreover, the BS broadcasts the value of ξ and UEs report their computation demands in terms of CPU clock frequency f_k^{c}. The remaining challenge now is to solve small-scale non-convex MINLP subproblems $(\mathcal{P}_3)_k$ to obtain the global optimum solution. Problem $(\mathcal{P}_3)_k$ is still difficult to tackle because it is a non-convex MINLP problem. Fortunately, the number of parallel tasks of each UE is not large in practice, for example 2 tasks in a face recognition application [95]; therefore, we can solve p_k and f_k^{c} by exploring all possible sets of $\boldsymbol{s}_k$ (i.e., all possible offloading solutions of UE k). Specifically, for each set of $\boldsymbol{s}_k$ satisfying $(5.22)_k$, let $\xi^{\mathsf{a}} = (w_k b_k^{\mathsf{a}})^{-1} W(\xi - w_k \xi_k^{\mathsf{lo}})$, $c_k^{\mathsf{a}} = \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) c_{k,l_k}$, and $b_k^{\mathsf{a}} = \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}$, we need to solve the following subproblem:

$$
\begin{aligned}
(\mathcal{P}_3)'_{\boldsymbol{s}_k} \quad & \min_{f_k^{\mathsf{c}}, p_k} \quad [f_k^{\mathsf{c}}]^+ \\
& \text{s. t.} \quad p_k + p_{k,0} - \xi^{\mathsf{a}} \log_2(1 + p_k \beta_k^{\mathsf{a}}) \leq 0, && \text{(5.26a)} \\
& \qquad \frac{b_k^{\mathsf{a}}}{W \log_2(1 + p_k \beta_k^{\mathsf{a}})} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} - \eta_k \leq 0, && \text{(5.26b)} \\
& \qquad (5.19g)_k,
\end{aligned}
$$

The optimal objective value of $(\mathcal{P}_3)_k$ is equal to the minimum of the optimal objective values of $(\mathcal{P}_3)'_{\boldsymbol{s}_k}$ considering all different combinations of s_{k,l_k} ($l_k \in \mathcal{L}_k$). The optimal solution structure for $(\mathcal{P}_3)'_{\boldsymbol{s}_k}$ is presented in the following proposition.

Algorithm 5.2. Solving Problem $(\mathcal{P}_3)'_{\boldsymbol{s}_k}$ for Set $\boldsymbol{s}_k$

Set $feasibility = true$ and compute b_k^{a}, ξ^{a} and c_k^{a}
Calculate $fg1 = \mathbb{1}_{b_k^{\mathsf{a}}/\eta_k - W\log_2(1+p_k\beta_k^{\mathsf{a}})<0}$.
if $fg1 = 1$ **then**
  Calculate $fg2 = \mathbb{1}_{g_k(p_k)\leq 0}$.
  if $fg2 = 1$ **then**
    Assign $p_k^\star{=}p_k$ and $(f_{k,\xi}^{\mathsf{c,min}})^{(\boldsymbol{s}_k)}$ as in (5.27).
  else
    Calculate $fg3 = \mathbb{1}_{g_k(p_k^{\mathsf{b}})\leq 0}$
    if $fg3 = 1$ **then**
      Calculate $fg4 = \mathbb{1}_{p_k^{\mathsf{a}}-p_k^{\mathsf{b}}\geq 0}.\mathbb{1}_{g_k(p_k^{\mathsf{a}})>0}$
      if $fg4 = 1$ **then**
        Assign $feasibility = false$
      else
        Apply (5.28) to obtain $p_k^\star$.
      end if
    else
      Assign $feasibility = false$
    end if
    if $feasibility$ **then**
      Assign $(f_{k,\xi}^{\mathsf{c,min}})^{(\boldsymbol{s}_k)}$ as in (5.27).
    end if
  end if
else
  Assign $feasibility = false$
end if

Proposition 5.3. *The solution of $(\mathcal{P}_3)'_{\boldsymbol{s}_k}$ corresponding to a set $\boldsymbol{s}_k$ can be expressed as*

$$(f_{k,\xi}^{\mathsf{c,min}})^{(\boldsymbol{s}_k)}{=}\frac{c_k^{\mathsf{a}}\log_2(1+p_k^\star\beta_k^{\mathsf{a}})}{\eta_k\log_2(1+p_k^\star\beta_k^{\mathsf{a}})-b_k^{\mathsf{a}}/W}, \tag{5.27}$$

where $p_k^\star = p_k$ if $g_k(p_k) \leq 0$ (note that $g_k(p_k) = p_k{+}p_{k,0}{-}\xi^{\mathsf{a}}\log_2(1{+}p_k\beta_k^{\mathsf{a}})$ defines the left-hand-side of constraints (5.26a); otherwise, $p_k^\star$ is the root of equation $g_k(p_k) = 0$ and it satisfies $p_k^{\mathsf{a}} < p_k^\star < p_k$ where $p_k^{\mathsf{a}} = (\beta_k^{\mathsf{a}})^{-1}(2^{b_k^{\mathsf{a}}/(\eta_k W)} - 1)$.

Proof. The proof is given in Appendix 5.9.1. □

Proposition 5.3 implies that we need to solve equation $g_k(p_k^\star) = 0$ to obtain the optimal solution if $g_k(p_k) > 0$. Toward this end, it is necessary to check whether or not there exists a value of $p_k^\star$ satisfying $p_k^{\mathsf{a}} < p_k^\star < p_k$ and $g_k(p_k^\star) = 0$. Because of the convexity of $g_k(p_k)$, the requisite conditions to ensure $(\mathrm{C0}')_k$ holds are $p_k^{\mathsf{b}} \leq p_k^\star < p_k$ and $g_k(p_k^{\mathsf{b}}) \leq 0$ where p_k^{b} is the stationary point

of $g_k(p_k)$ (the point for which the first derivative of $g_k(p_k)$ equals to zero), which can be derived as $p_k^{\mathsf{b}} = \frac{\xi^{\mathsf{a}}}{\ln 2} - \frac{1}{\beta_k^{\mathsf{a}}}$. If $p_k^{\mathsf{b}} \leq p_k^{\mathsf{a}}$ and $g_k(p_k^{\mathsf{a}}) > 0$, then $g_k(p_k) > 0$ for $p_k^{\mathsf{a}} < p_k < p_k$. If all requisite conditions are satisfied, we can find the root $p_k^{\star} \geq p_k^{\mathsf{b}}$ by employing the Newton-Raphson search method through the following iterative updates:

$$p_k^{t+1} = p_k^{\mathsf{t}} - \frac{g_k(p_k^{\mathsf{t}})(1 + p_k^{\mathsf{t}}\beta_k^{\mathsf{a}})\ln(2)}{(1 + p_k^{\mathsf{t}}\beta_k^{\mathsf{a}})\ln(2) - \xi^{\mathsf{a}}\beta_k^{\mathsf{a}}}, \tag{5.28}$$

where the initial point p_k^0 can be set equal to p_k. Note that we can also apply one dimensional bisection search to find the largest possible value of p_k.

In summary, the optimal solution of $(\mathcal{P}_3)'_k$ can be obtained as described in Algorithm 5.2 where $\mathbb{1}_x$ denotes an indicator function (i.e., $\mathbb{1}_x = 1$ if condition x holds and $\mathbb{1}_x = 0$, otherwise). Moreover, $(\mathcal{P}_3)_k$ is feasible if there exists at least one set $\boldsymbol{s}_k$ so that $(\mathcal{P}_3)'_{\boldsymbol{s}_k}$ is feasible and we then have $f_{k,\xi}^{\mathsf{c,min}} = \min_{\boldsymbol{s}_k}(f_{k,\xi}^{\mathsf{c,min}})^{(\boldsymbol{s}_k)}$.

5.4.2 P-CSI - Low-complexity Algorithm (P-SO)

The complexity of the optimal algorithm (i.e., Algorithm 5.1) strongly depends on the number of possible combinations of task offloading decisions. Thus, it can be highly complex to find the optimal solution when the number of computation tasks is large. To deal with such complexity, we propose a low-complexity algorithm which iteratively solves two subproblems decomposed from problem $(\mathcal{P}_2)$ where the first one, i.e., the offloading optimization (OP) subproblem, determines offloading decision and computing resource allocation while the second one, i.e., the power allocation (PA) subproblem, performs uplink power allocation and reassigns the computing resource. First, for a given value of $\boldsymbol{p}$, the (OP) subproblem is given as follows:

$$(\mathcal{P}_2^{\mathsf{OP}}) \min_{\boldsymbol{S},\boldsymbol{f}^{\mathsf{c}},\xi} \xi \quad \text{s.t} \quad (5.20b), (5.19c) - (5.19e), (5.22).$$

Second, with the offloading solution $\boldsymbol{S}$ obtained by solving $(\mathcal{P}_2^{\mathsf{OP}})$, the (PA) is given as

$$(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}) \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\xi} \xi \quad \text{s.t} \quad (5.26a), (5.19c), (5.19e), (5.19g).$$

The proposed algorithm (P-SO), which iteratively solves $(\mathcal{P}_2^{\mathsf{OP}})$ and $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ until convergence, is described in Algorithm 5.3. Besides, this approach is the key for solving problem $(\mathcal{P}_2)$ in the IP-CSI scenario when the finding of optimal solution would be impossible. Note that $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$ is the optimal of subproblem $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ at iteration q. We describe how to solve (OP) and (PA) in the following.

5.4.2.1 Offloading Subproblem (OP)

In order to tackle this subproblem, we will apply the decomposition technique as employed in section 5.4.1 to further decompose this subproblem into individual users' small-scale subproblems:

$$(\mathcal{P}_2^{\mathsf{OP}})_k \quad \min_{\boldsymbol{s}_k, f_k^{\mathsf{c}}} \quad [f_k^{\mathsf{c}}]^+ \text{s.t.} \quad (5.20b)_k, (5.19c)_k, (5.19d)_k, (5.22)_k.$$

To solve $(\mathcal{P}_2^{\mathsf{OP}})_k$, we propose two methods: the first one will transform the non-convex constraints into convex constraints and then solve the corresponding convex MINLP (**Method 1**) while the other will find $f_{k,\xi}^{\mathsf{c,min}}$ by directly dealing with all possible combinations of $\boldsymbol{s}_k$ (**Method 2**).

a) Method 1: We first rewrite the offloading time constraint $(5.19c)_k$ as

$$r_k^{-1} \sum_{l_k \in \mathcal{L}_k} f_k^{\mathsf{c}}(1 - s_{k,l_k}) b_{k,l_k} + \sum_{l_k \in \mathcal{L}_k} (1 - s_{k,l_k}) c_{k,l_k} - f_k^{\mathsf{c}} \eta_k \leq 0. \tag{5.29}$$

The non-convex term $z_{k,l_k} = f_k^{\mathsf{c}} s_{k,l_k}$ can be transformed into a linear form as follows:

$$0 \leq z_{k,l_k} \leq s_{k,l_k} F^{\mathsf{c}}, \tag{5.30}$$

$$0 \leq f_k^{\mathsf{c}} - z_{k,l_k} \leq (1 - s_{k,l_k}) F^{\mathsf{c}}. \tag{5.31}$$

The small-scale subproblem $(\mathcal{P}_2^{\mathsf{OP}})_k$ with these transformations becomes a convex mixed integer non-linear (cubic polynomial) problem of $\xi, \boldsymbol{s}_k, \boldsymbol{z}_k, f_k^{\mathsf{c}}$, which can be solved efficiently by available solvers such as GAMS-BARON, CVX-Gurobi, CVX-MOSEK thanks to the convexity property [118].

b) Method 2: In this method, we apply bisection search on ξ as in Algorithm 5.1, except for some difference in step 4 and step 5 to find $f_{k,\xi}^{\mathsf{c,min}}$. Let $\boldsymbol{S}_k^{\mathsf{bi}} \in \mathbb{R}^{2^{|\mathcal{L}_k|} \times |\mathcal{L}_k|}$ denote the binary matrix

Algorithm 5.3. Low-complexity Algorithm - P-CSI (P-SO)

1: **Initialize**: choose ϵ, initial $p_k^{(0)} = p_{\max}/2, \forall k$.
2: **while** $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})} - \xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{PA}})} < \epsilon$ **do**
3: Assign $q = q + 1$;
4: Solve $\mathcal{P}_2^{\mathsf{OP}}$ to get $\boldsymbol{S}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})}$.
5: Solve $\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}}$ to get $\boldsymbol{p}^{(q)}$, $(\boldsymbol{f}^{\mathsf{c}})^{(q)}$ and $\xi^{(q)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})}$.
6: **end while**

whose rows represent all possible combinations of task offloading decisions of UE k. For example, $\boldsymbol{S}_k^{\mathsf{bi}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}^{\mathsf{t}}$ in case of $|\mathcal{L}_k| = 2$. Then, it can be verified that the minimum value of f_k^{c} for a given value of ξ can be computed as

$$\begin{aligned} f_{k,\xi}^{\mathsf{c,min}} &= \min((LC0_k \odot LC2_k \odot L20_k)\backslash\{0\}), \\ LC0_k &= \mathbb{1}_{\xi \times \mathbf{1}_{2^{|\mathcal{L}_k|}} - LC0'_k \geq 0}, \\ LC0'_k &= w_k \alpha_k \eta_k^{-2} (\boldsymbol{S}_k^{\mathsf{bi}} \boldsymbol{c}_k)^3 + \frac{w_k (p_k + p_{k,0})(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{b}_k}{r_k^{\mathsf{lb}}}, \\ LC2_k &= \left[(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{c}_k \oslash (\eta_k \times \mathbf{1}_{2^{|\mathcal{L}_k|}} - \frac{(1 - \boldsymbol{S}_k^{\mathsf{bi}}) \boldsymbol{b}_k}{r_k^{\mathsf{lb}}}) \right]^+, \\ L20_k &= \mathbb{1}_{\eta_k F_k - \boldsymbol{S}_k^{\mathsf{bi}} \boldsymbol{c}_k \geq 0}, \end{aligned} \tag{5.32}$$

where $\odot$ and $\oslash$ denote the Hadamard product and division, respectively, $\mathbf{1}_n$ represents the $n \times 1$ vector of ones, $\mathbb{1}_{\boldsymbol{x} \geq 0}$ is the indicator function and $\boldsymbol{x}^+ = \max(\boldsymbol{x}, 0)$. In above expressions, the elements of $LC0_k$ and $L20_k$ will be equal to 1 if the corresponding row of $\boldsymbol{S}_k^{\mathsf{bi}}$ satisfies constraint $(5.20b)_k$ and $(5.22)_k$, respectively. The vector of $LC2_k$ describes the minimum value of f_k^{c} corresponding to each row of $\boldsymbol{S}_k^{\mathsf{bi}}$.

It is noted that $LC0'_k, LC2_k$ and $L20_k$ do not depend on the value of ξ; thus, we just need to compute them at the beginning of the bisection search (the 'while-loop' in Algorithm 5.1) and use them to update $LC0_k$ and $f_{k,\xi}^{\mathsf{c,min}}$ corresponding to the updated value of ξ. Because this method considers all possible values of $\boldsymbol{s}_k$, it is more suitable for the setting with a small number of tasks per UE.

5.4.2.2 Uplink Power Allocation Subproblem (PA)

With the solution of $(\mathcal{P}_2^{\mathsf{OP}})$, we can then solve the $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ subproblem to obtain the optimal solutions of transmit power $\boldsymbol{p}_k$ and computing resource allocation $\boldsymbol{f}^{\mathsf{c}}$. This can be fulfilled by using a similar process employed in section 5.4.1 which applies the bisection search on ξ and solving subproblem $(\mathcal{P}_3)'_k$. We state important results for Algorithm 5.3 in the following proposition.

Proposition 5.4. *Algorithm 5.3 creates a sequence of feasible solutions for* $(\mathcal{P}_2)$ *where objective function value of this problem monotonically decreases over iterations.*

Proof. Let $(\mathcal{V}^{\mathsf{OP}})^{(q-1)}$ denote the optimal point of $(\mathcal{P}_2^{\mathsf{OP}})$ and $(\mathcal{V}^{\mathsf{PA}})^{(q-1)}$ denote the optimal point of $(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})$ at iteration $q-1$. Clearly, $((\mathcal{V}^{\mathsf{PA}})^{(q-1)}, \boldsymbol{S}^{(q-1)})$ is a feasible solution of $(\mathcal{P}_2^{\mathsf{OP}})$ at iteration q. Thus, at iteration q, we have $\xi^{(q)}|_{(\mathcal{P}_2^{\mathsf{OP}})} = \min\limits_{\mathcal{V}^{\mathsf{OP}} \supset \{(\mathcal{V}^{\mathsf{PA}})^{(q-1)}, \boldsymbol{S}^{(q-1)}\}} \xi|_{(\mathcal{P}_2^{\mathsf{OP}})} \leq \xi^{(q-1)}|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})} = \min\limits_{\mathcal{V}^{\mathsf{PA}} \supset \{(\mathcal{V}^{\mathsf{OP}})^{(q-1)}, \boldsymbol{p}^{(q-2)}\}} \xi|_{(\mathcal{P}_{2,\mathsf{P}}^{\mathsf{PA}})} \leq \xi^{(q-1)}|_{(\mathcal{P}_2^{\mathsf{OP}})}$. Hence, the iterative process will converge in a finite number of iterations. □

5.5 Algorithm Design for IP-CSI Scenario

We employ a similar approach, which is used to develop the low-complexity algorithm (P-SO) for the P-CSI scenario to tackle the problem in this IP-CSI scenario. Specifically, we tackle problem $(\mathcal{P}_2)$ by iteratively solving two subproblems $(\mathcal{P}_2^{\mathsf{OP}})$ and $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ until convergence. The proposed algorithm for the IP-CSI scenario is refered to as (IP-SO) in the sequel. This (IP-SO) algorithm is similar to Algorithm 5.3; hence, we do not present it for brevity. Because the IP-CSI only affects the transmission energy and transmission time, the (OP) subproblem $(\mathcal{P}_2^{\mathsf{OP}})$ can be solved as in Section 5.4.2.1. We only need to consider the (PA) subproblem $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$, which can be written as

$$
\begin{aligned}
(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}}) \quad & \min_{\boldsymbol{p}, \boldsymbol{f}^{\mathsf{c}}} \quad \xi \\
& \text{s. t.} \quad (5.20b): \ w_k(\xi_{k,\mathsf{IP}}^{\mathsf{t,ub}} + \xi_k^{\mathsf{lo}}) \leq \xi, \\
& \qquad\quad (5.19c): \ t_{k,\mathsf{IP}}^{\mathsf{t,ub}} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} \leq \eta_k, \ \ (5.19e), (5.19g).
\end{aligned}
$$

Assuming that the UE's training power is fixed in the CSI estimation phase, the NP-hardness of $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ is stated in the following proposition.

Proposition 5.5. *The subproblem* $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ *is NP-hard.*

Proof. The proof is given in Appendix 5.9.2. □

Even though $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ is non-convex and NP-hard, we can solve it by using the bisection search method. The key step in this bisection search is to perform the feasibility verification of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ for a given ξ (as can be seen in Algorithm 5.1). We propose to employ the DC optimization method to convexify and tackle this feasibility verification problem [93]. We will show later that our proposed algorithm to solve $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ in this IP-CSI scenario converges to a stationary point, which, therefore, guarantees the convergence of the main algorithm (IP-SO) as stated in Proposition 5.4.

We now proceed to address the feasibility verification problem for $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$. The transmission rate in (16) is the difference between two concave functions, which makes $1/\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$ in constraints (5.20b) and (5.19c) non-convex. To convexify $1/\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$, we approximate $\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$ at point $\boldsymbol{p}^{(q)}$ by a concave function $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$. Indeed, it can be verified that $1/x$ is a convex and non-increasing function of x and if $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ is a concave function of $\boldsymbol{p}$, the composition function $1/\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ is convex in $\boldsymbol{p}$ [1].

To obtain a concave function $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$, the second concave term $v_k(\boldsymbol{p}) = W\log_2\left(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\right)$ at point $\boldsymbol{p}^{(q)}$ of $\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$ in (5.16) can be approximated by a linear function as

$$v_k(\boldsymbol{p}) \leq \tilde{v}_k(\boldsymbol{p}) = v_k(\boldsymbol{p}^{(q)}) + \nabla v_k(\boldsymbol{p}^{(q)})(\boldsymbol{p} - \boldsymbol{p}^{(q)}), \tag{5.33}$$

where $\nabla v_k(\boldsymbol{p}^{(q)})$ is the gradient of v_k at point $\boldsymbol{p}^{(q)}$. Using this approximation, for a given value of ξ and point $\boldsymbol{p}^{(q)}$, we can obtain the concave approximation function $\tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ of $\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p})$ by using its lower bound as follows:

$$\hat{r}^{\mathsf{lb}}_k(\boldsymbol{p}) \geq W\log_2(p_k + \boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - W\log_2((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k) - \frac{\boldsymbol{\lambda}_k(\boldsymbol{p} - \boldsymbol{p}^{(q)})}{\log(2)\left((\boldsymbol{p}^{(q)})^{\mathsf{t}}\boldsymbol{\lambda}_k + \sigma_k\right)} = \tilde{r}^{\mathsf{lb}}_k(\boldsymbol{p}|\boldsymbol{p}^{(q)}). \tag{5.34}$$

As a result, for a given value of ξ and point $\boldsymbol{p}^{(q)}$, constraints (5.20b) and (5.19c) can be approximated by the following constraints, respectively:

$$p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}} \tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq 0, \tag{5.35}$$

$$\left(\frac{T}{T-\tau}\right) \frac{b_k^{\mathsf{a}}}{\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} - \eta_k \leq 0. \tag{5.36}$$

From (5.34), the feasibility verification of $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ for a given value of ξ is now equivalent to find at least one point $(\boldsymbol{f}^{\mathsf{c}}, \boldsymbol{p}, \boldsymbol{p}^{(q)})$ that makes constraints (5.19e) and (5.19g) and inequalities (5.35), (5.36) feasible. Toward this end, we will iteratively update $\boldsymbol{p}^{(q)}$ to make the approximation in (5.34) tighter and find the minimum χ for a given $\boldsymbol{p}^{(q)}$, where χ is an upper-bound of the functions in the left-hand-side of (5.35), (5.36) for all users and for all $\boldsymbol{p}, \boldsymbol{f}^{\mathsf{c}}$ satisfying (5.19e) and (5.19g). It is clear that if we can find such a minimum $\chi \leq 0$, then constraints (5.35), (5.36) are satisfied; therefore, $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ will be feasible. Specifically, the minimum value of χ for a given $\boldsymbol{p}^{(q)}$ can be found by solving the following problem:

$$(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})^{(q)} \quad \min_{\boldsymbol{p},\boldsymbol{f}^{\mathsf{c}},\chi} \quad \chi \tag{5.37a}$$

$$\text{s. t.} \quad p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}} + p_{k,0})}{T-\tau} - \xi^{\mathsf{a}} \tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq \chi, \tag{5.37b}$$

$$\left(\frac{T}{T-\tau}\right) \frac{b_k^{\mathsf{a}}}{\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} - \eta_k \leq \chi, \tag{5.37c}$$

$$(5.19e), (5.19g),$$

As shown above, the convexity of $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})^{(q)}$ is guaranteed; therefore, we can effectively solve this problem by using the CVX solver. Finally, the feasibility verification is presented in Algorithm 5.4, and the bisection search to solve $(\mathcal{P}_{2,\mathrm{IP}}^{\mathsf{PA}})$ is similar to Algorithm 5.1, except for the difference in step 5, where the feasibility verification is done as described in Algorithm 5.4.

We will show that our proposed algorithm to solve $(\mathcal{P}_{2,\mathrm{IP}}^{\mathsf{PA}})$ converges to a stationary point, which is stated in the following Proposition 5.6 and Proposition 5.7.

Proposition 5.6. *For a given ξ, using D.C to approximate the transmission rate (using the rate lower bound in (5.34)) and iteratively solving problem $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})^{(q)}$ leads to convergence.*

Algorithm 5.4. PA Feasibility Verification - IP-CSI

Initialize: choose $\boldsymbol{p}^{(0)}$ as the previous solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$.
repeat
  $q = q + 1$;
  At $\boldsymbol{p} = \boldsymbol{p}^{(q-1)}$, solve $(\mathcal{P}^{\mathsf{PA}}_{2})^{(q-1)}$ to get $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$
  if $\chi < 0$ **then**
    Assign $feasibility = true$
    Return $\boldsymbol{p}^{(q)}$, $\boldsymbol{f}^{\mathsf{c}}$; **break**;
  else
    Assign $feasibility = false$
    Compute $\tilde{r}^{\mathsf{lb}}_{k}(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ for all k
  end if
until convergence

Proof. The proof is given in Appendix 5.9.3. □

Proposition 5.7. *If the optimal value of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ is equal to zero at the convergence of Algorithm 5.4, then the solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ combining with ξ gives a stationary point of subproblem $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$, where Q denotes the final iteration index at the convergence of Algorithm 5.4.*

Proof. When the optimal value of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ is equal to zero at the convergence, the bisection search of ξ will terminate and the solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ combining with ξ is the solution of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$. On the other hand, the Karush-Kuhn-Tucker (KKT) conditions of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$ and $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ at convergence are the same. Moreover, in each iteration of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})^{(Q)}$ we always obtain the optimal solution, which therefore satisfies the KKT conditions of $(\mathcal{P}^{\mathsf{PA}}_{2,\mathsf{IP}})$. □

5.6 Extension and Complexity Analysis

5.6.1 Consideration of Downlink Transmission

For certain applications such as virtual-reality games, the amount of downlink data and time to send back the computation result from the BS can be non-negligible and it must be taken into account in the computation offloading design. In this section, we extend the proposed design described in the previous sections to consider this downlink transmission.

Let b_{k,l_k}^{dl} denote the number of downlink bits related to the computation result of task l_k, which must be sent from the BS to UE k. We assume the time division duplexing (TDD) wireless system[1], the downllink combining can be realized by using the same estimated channel matrix $\hat{\boldsymbol{H}}$ in the uplink. Let p_k^{dl} denote the power that BS uses to transmit the offloading result data to UE k, then the downlink signal-to-noise-plus-interference ratio of UE k is given by

$$\gamma_{k,\mathsf{dl}} = \frac{p_k^{\mathsf{dl}} |\hat{\boldsymbol{h}}_k^H \hat{\boldsymbol{a}}_k|^2}{\sum\limits_{i\neq k, i\in\mathcal{K}_\xi} p_i^{\mathsf{dl}} |\hat{\boldsymbol{h}}_k^H \hat{\boldsymbol{a}}_i|^2 + \frac{\sigma_{bs}\beta_k}{\tau p^{\mathsf{tr}}\beta_k + \sigma_{bs}} \sum\limits_{i\in\mathcal{K}_\xi} p_i^{\mathsf{dl}} |\hat{\boldsymbol{a}}_i|^2 + \sigma_k^{\mathsf{dl}}}, \tag{5.38}$$

where σ_k^{dl} represents the received noise power at UE k. In the P-CSI scenario, the error in the second term of the denominator is zero so this term does not exist. Assuming that ZF precoder is employed, the lower-bound ergodic downlink rate can be espressed as

$$\hat{r}_k^{\mathsf{dl,lb}} = \begin{cases} W \log_2\left(1 + p_k^{\mathsf{dl}}/\sigma_k^{\mathsf{dl}}\right) & \text{(P-CSI)} \\ W \log_2\left(1 + \frac{p_k^{\mathsf{dl}}}{\sum\limits_{i\in\mathcal{K}_\xi} p_i^{\mathsf{dl}} \lambda_{k,i} + \sigma_k^{\mathsf{dl}}}\right) & \text{(IP-CSI)} \end{cases}. \tag{5.39}$$

Then, the constraint (5.19c) on the total latency, which includes the computation time, the upload and download time, can be expressed as

$$\frac{T}{T-\tau}\left(\frac{b_k^{\mathsf{a}}}{\hat{r}_k^{\mathsf{lb}}} + \frac{b_k^{\mathsf{a,dl}}}{\hat{r}_k^{\mathsf{dl,lb}}}\right) + \frac{c_k^{\mathsf{a}}}{f_k^c} \leq \eta_k, \quad \forall k \in \mathcal{K}_\xi, \tag{5.40}$$

where $b_k^{\mathsf{a,dl}} = \sum\limits_{l_k\in\mathcal{L}_k} (1 - s_{k,l_k}) b_{k,l_k}^{\mathsf{dl}}$.

On the other hand, the average transmit power of BS can be computed as

$$p_{\mathsf{BS}} = \sum_{k\in\mathcal{K}_\xi} p_k^{\mathsf{dl}} \mathbb{E}(||\hat{\boldsymbol{a}}_{\boldsymbol{k}}||^{\mathbf{2}}) = \begin{cases} \sum\limits_{k\in\mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}}{(M - |\mathcal{K}|_\xi)\beta_k}, \text{ P-CSI} \\ \sum\limits_{k\in\mathcal{K}_\xi} \frac{p_k^{\mathsf{dl}}(\tau p^{\mathsf{tr}}\beta_k + \sigma_{bs})}{(M - |\mathcal{K}|_\xi)\tau p^{\mathsf{tr}}\beta_k^2}, \text{ IP-CSI} \end{cases}.$$

[1]The TDD approach has been advocated recently because it can significantly reduce the CSI estimation overhead, especially in massive MIMO wireless systems.

The total transmit power at BS must be constrained by its maximum power $p_{\mathsf{max}}^{\mathsf{dl}}$, which can be expressed as

$$p_{\mathsf{BS}} \leq p_{\mathsf{max}}^{\mathsf{dl}}. \tag{5.41}$$

We can now formulate the joint computation offloading and resource allocation problem considering both uplink and downlink data transmissions as follows:

$$(\mathcal{P}_2^{\mathsf{ext}}) \ \min \xi \quad \text{s.t} \quad (5.20b), (5.19d), (5.19e), (5.19g), (5.22), (5.40), (5.41).$$

The difference between $(\mathcal{P}_2)$ and $(\mathcal{P}_2^{\mathsf{ext}})$ is in constraints (5.40) and (5.41). To tackle this difficult problem, we can again decompose it into two subproblems as in previous sections. In particular, we iteratively solve the (OP) subproblem (with constraints (5.20b), (5.19d), (5.19e), (5.22), (5.40)) to find the optimal $\boldsymbol{s}, \boldsymbol{f}^{\mathsf{c}}$ and solve the extended (PA) subproblem (with constraints (5.20b), (5.19e), (5.19g), (5.40), (5.41)) to find $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$. Because the optimization variables $\boldsymbol{p}, \boldsymbol{p}^{\mathsf{dl}}$ are only captured in the extended (PA) subproblem, we can solve the (OP) subproblem as in section 5.4.2.1. We now discuss how to solve the extended (PA) subproblem which is stated as follows:

$$(\mathcal{P}_2^{\mathsf{PA,ext}}) \ \min \xi \quad \text{s.t} \quad (5.20b), (5.19e), (5.19g), (5.40), (5.41).$$

Considering the first delay term in (5.40), the two delay components, which correspond to the uplink transmission time of the incurred data and download transmission time of the computation outcome, respectively, have the same structure. Therefore, we can apply the same techniques as in the previous sections to deal with the downlink rate.

Specifically, we will apply the bisection search to find ξ for which we have to perform feasibility verification for a given value of ξ (to update the upper and lower bounds of ξ). For the P-CSI scenario, the non-convex constraint (5.20b) can be convexified by rewriting it as (5.26a); therefore, the feasibility verification can be completed by using the CVX solver. For the IP-CSI scenario, to solve the (PA) subproblem, we employ the DC optimization technique as in Section 5.5 to deal with non-convex constraints involving both uplink rate and downlink rate.

5.6.2 Complexity Analysis

We analyze the computational complexity of the proposed algorithms in term of the number of required arithmetic operations. In Algorithm 5.1, the main complexity comes from the while-loop for the bisection search and the process of solving subproblem $(\mathcal{P}_3)_k$ in step 4. The bisection search of ξ requires $\log_2(\frac{\xi_{\max}-\xi_{\min}}{\epsilon})$ iterations. Besides, the Newton-Raphson search method to solve $g_k(p_k^\star) = 0$ typically converges within tens of iterations, denoted by N_1, and each iteration has complexity of $\mathcal{O}(1)$. Therefore, the computational complexity involved in solving subproblem $(\mathcal{P}_3)_k$ is $\mathcal{O}(2^{|\mathcal{L}_k|}N_1)$. Thus, the overall complexity of Algorithm 5.1 is $\mathcal{O}(\log_2(\frac{\xi_{\max}-\xi_{\min}}{\epsilon})2^{|\mathcal{L}_k^{\mathsf{m}}|}N_1K)$, where $|\mathcal{L}_k^{\mathsf{m}}| = \max_k |\mathcal{L}_k|$.

For the (P-SO) algorithm (Algorithm 5.3) described in Section 5.4.2, we describe the worst-case complexity with exhaustive search using Method 2 to solve (OP) subproblem. The computational complexity involved in solving (OP) is $\mathcal{O}(2^{|\mathcal{L}_k^{\mathsf{m}}|}K)$ because the exhaustive search using Method 2 just needs to compute at the beginning of the bisection search. Therefore, the worst-case complexity of Algorithm 5.3 is $\mathcal{O}(N_2(2^{|\mathcal{L}_k^{\mathsf{m}}|}K + \log_2(\frac{\xi_{\max}-\xi_{\min}}{\epsilon})N_1K))$, where N_2 denotes the number of iterations required by two subproblems (OP) and (PA) to achieve convergence. As shown in the simulation result later, N_2 is typically no more than 6.[2]

For the IP-CSI scenario, the iterative process in Algorithm 5.4 converges in a few iterations, which is denoted as N_3. In each iteration, the convex problem $(\mathcal{P}_2^{\mathsf{PA}})^{(q)}$ and $(\mathcal{P}_2^{\mathsf{PA,ext}})^{(q)}$ can be solved by using the interior-point method with complexity $\mathcal{O}(m^{1/2}(m+n)n^2)$, where m denotes the number of inequality constraints, and n represents the number of variables [119]. Therefore, the complexity of solving $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA}})$ and $(\mathcal{P}_{2,\mathsf{IP}}^{\mathsf{PA,ext}})$ are similar, which is equal to $\mathcal{O}(K^{3.5}N_3)$. Consequently, the overall complexity of solving $(\mathcal{P}_2)$ and $(\mathcal{P}_2^{\mathsf{ext}})$ in the IP-CSI scenario is $\mathcal{O}(N_2(2^{|\mathcal{L}_k^{\mathsf{m}}|}K + \log_2(\frac{\xi_{\max}-\xi_{\min}}{\epsilon})K^{3.5}N_3))$.

[2]The complexity involved in solving the relaxed convex problem of $(\mathcal{P}_2^{\mathsf{OP}})_k$ is $\mathcal{O}\left(|\mathcal{L}_k|^{3.5}\right)$ and our numerical studies suggest that the **average complexity** required to solve the original MINLP $(\mathcal{P}_2^{\mathsf{OP}})_k$ is approximately $\mathcal{O}\left(|\mathcal{L}_k|^{6.5}\right)$, which is much smaller than its worst-case complexity.

5.7 Numerical Results

We consider an MEC system with the channel bandwidth of 10 MHz and $K = 20$ UEs randomly distributed in a cell coverage area with the radius of 900m. All UEs are assumed to have the same maximum clock frequency of 2.4 GHz, the same maximum transmit power (i.e., $p_k = p_{\max}$), which is set equal to 0.22 (Watts) according to the 3GPP technical report [120] and the circuit power is set equal to 0.05 (Watts). In our simulation setting, all UEs have the same number of parallel tasks and the same total computation demand of 0.24 Gcycles, but the number of CPU cycles per task is set randomly and uniformly. The total number of transmission bits for all tasks is set to be the same for all UEs while the number of bits per task is generated randomly and uniformly. For performance evaluation of the proposed design, we choose the ratio between the total number of transmission bits and the total required CPU cycles (BPC) to be about 4.2×10^{-3} (except for the results in Fig. 5.3 and Fig. 5.5), which is close to its highest possible value for the applications considered in [26].

The energy weights w_k are set equal to 1 for fair comparison with the no-computation-offload case. The energy coefficient is set as $\alpha_k = \alpha = 0.1\times10^{-27}$, which corresponds to the realistic measurement value in [99] and recent development in mobile chipset technology. The effect of this parameter on the performance of the offloading design will be clarified in Fig. 5.6. For all considered simulation scenarios, we set $|\mathcal{L}_k| = L = 5$ (except in Fig. 5.8), $M = 30$ (except in Fig. 5.1), $F^{\mathsf{c}} = 40$ GHz (except in Fig. 5.6), maximum allowable delay $\eta_k = \eta$ for all UEs, and $p^{\mathsf{dl}}_{\mathsf{max}} = 10$ (Watts).

The noise powers at the mobile user side and BS are given as $\sigma_{bs} = \sigma^{\mathsf{dl}}_k$ = bandwidth $\times k_B \times T_0 \times$ noise figure (W), where $k_B = 1.381 \times 10^{-23}$ (Joule per Kelvin) is the Boltzmann constant, $T_0 = 290$ (Kelvin) is the noise temperature, and noise figure = 0.9 (Watts). The small scale channel fading coefficient is generated according to the Rayleigh distribution and the path-loss is defined according to 3GPP technical report as β_k (dB) $= 128.1 + 37.6 \log_{10}(d_k)$ where $d_k > 0.01$ is the geographical distance between UE k and the BS (in km) [27]. In the IP-CSI scenario, we take $T = 200$ symbols, which corresponds to the coherence bandwidth of 200 kHz and a coherence time of 1 (ms). The simulation results are obtained by averaging the results over 100 realizations except for Fig. 5.1 and Fig. 5.4.

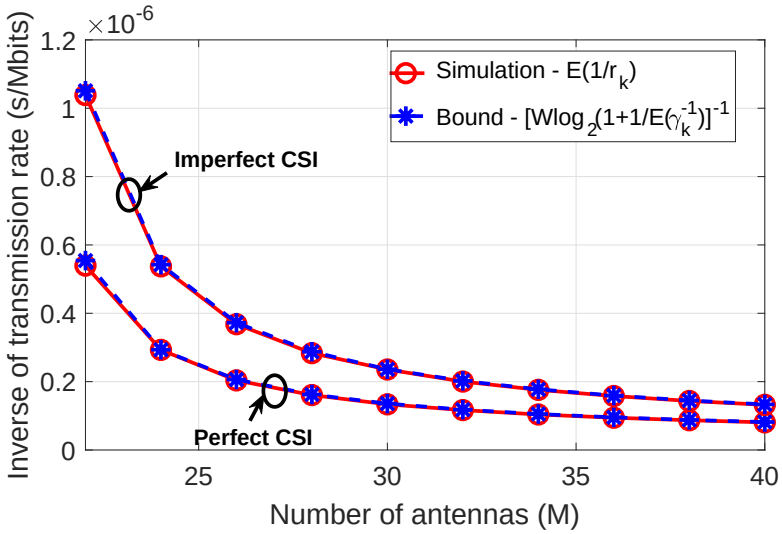

Figure 5.1 – Simulated and lower-bound of inverse rate.

Fig. 5.1 compares the upper-bound of the inverse of transmission rate given in the right-hand-side of (5.10) and its simulated values obtained by averaging over 10000 realizations in both P-CSI and IP-CSI scenarios when the transmit power $p_k = p_{\max}/2$ is set equally for all UEs. This figure confirms that the value of ergodic inverse of transmission rate is indeed close to its upper-bound and the gap between them is negligible when the number of antennas at the BS M becomes relatively large compared to the number of UEs K. In fact, when the SINR is good enough to support the data transmission required by the offloading process, as shown in (5.9), the second-order derivative of the inverse of the transmission rate is nearly equal to zero. Therefore, the equality condition of the Jensen's inequality in (5.10) holds with high probability. Another reason is the channel hardening property in massive MIMO, thus a fading channel behaves as if it was a non-fading channel.

The convergence of the proposed low-complexity algorithms based on bisection search for both P-CSI scenario (P-SO) and IP-CSI scenario with or without downlink transmission (IP-SO) is illustrated in Fig. 5.2. Specifically, we show the variations of the maximum W.C.E (ξ) over iterations in the left subfigure where no-downlink transmission is indicated as noDL and with-downlink transmission is shown together with the ratio between downlink data size and uplink data size $\Gamma_{\mathsf{dpu}} = 1$. The right subfigure shows the variations of χ in (5.37a) over iterations for different values of ξ, which is used to perform feasibility verification in the bisection search in the (PA) subproblem. It can be seen that χ converges to a negative value for some values of ξ, which indicates the feasibility condition.

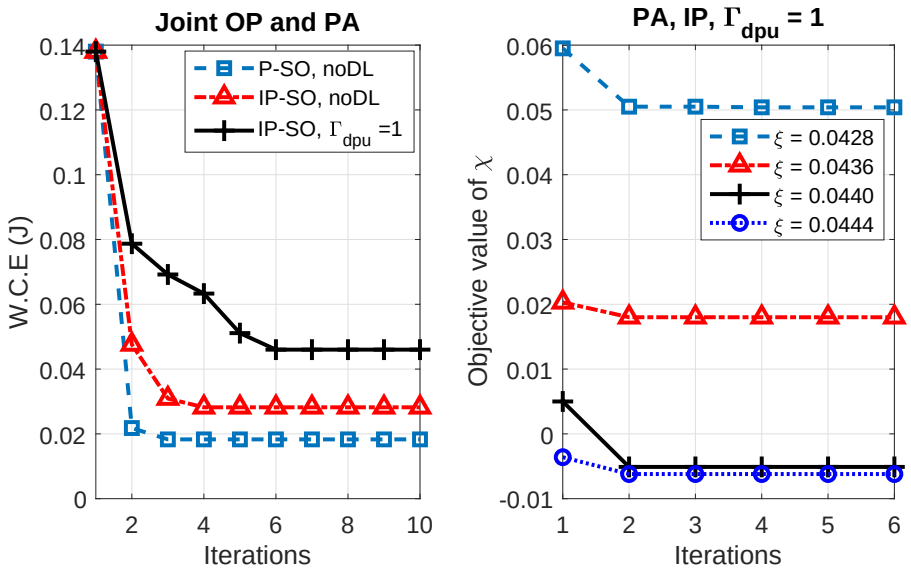

Figure 5.2 – Convergence of proposed algorithms.

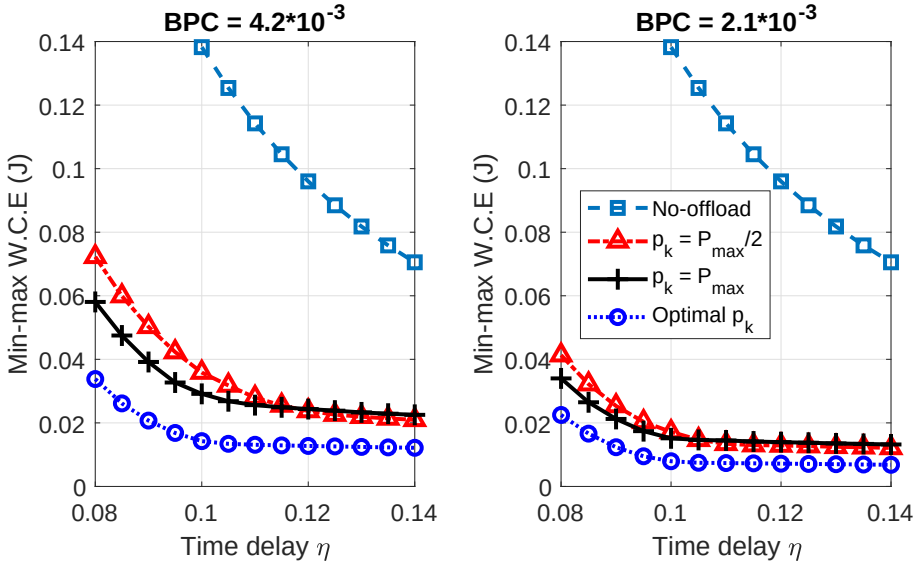

Figure 5.3 – Performance comparison of with/without offloading and with/without optimization of radio and computing resource.

The benefit of joint optimization of radio and computing resource allocation in the computation offloading design is illustrated in Fig. 5.3 for varying maximum allowable delay η. In this figure, considering no downlink data transmission and P-CSI scenario, we compare the achievable performance in four scenarios: task processing at mobile devices ('No-offload'), partial offloading with optimal offloading decision and cloud-resource allocation with fixed transmit power for all UEs $p_k = p_{\max}/2$ and $p_k = p_{\max}$, and with optimal transmit power allocation ('Optimal p_k'). The left and right subfigures show the achieved min-max W.C.E for different values of transmission bits per CPU cycle (BPC). From this figure, we can see that the minimum required latency that the mobile

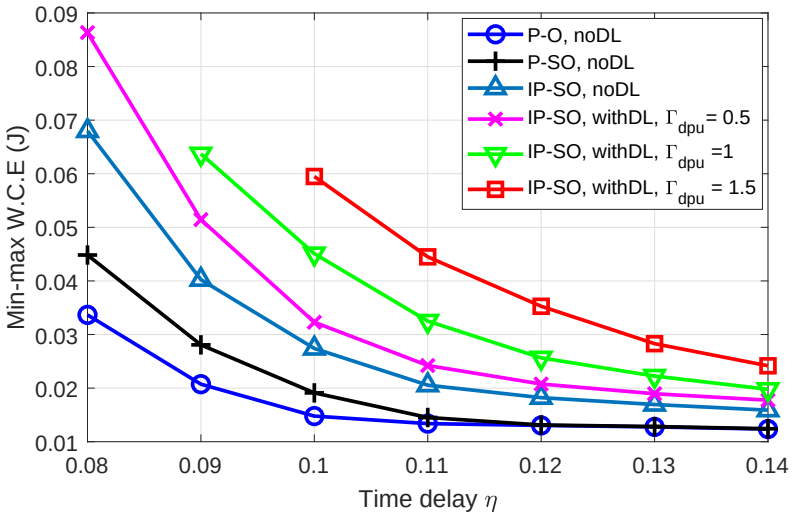

Figure 5.4 – Min-max W.C.E versus maximum allowable latency.

device can process its tasks locally ('No-offload' case) is 0.1s while the minimum required latency in the remaining cases are 0.08s. This means that computation offloading allows mobile devices to achieve lower latency. Moreover, the consumed energy in the partial offloading scheme is significantly smaller than that in the 'No-offload' case. For instance, the min-max W.C.E at $\eta = 0.1s$ in the left subfigure is equal to 0.138, 0.036, 0.029, 0.014 for the 'No-offload', fixed transmit power of '$p_k = p_{\max}/2$', '$p_k = p_{\max}$' and 'Optimal p_k', respectively. This means that partial offloading enables saving about 5 times of energy with no optimization of the transmit power and save about 10 times of energy with optimal transmit power. Moreover, the difference in the consumed energy among the offloading and no-offloading schemes becomes larger for smaller number of transmission bits.

Fig. 5.4 presents the achieved performance of different design scenarios considered in this paper: optimal solution with P-CSI - no downlink data ('P-O, noDL'), solution with P-CSI - no downlink data ('P-SO, noDL'), and IP-CSI - no downlink data ('IP-SO, noDL'). We also consider different application scenarios with small, medium and large amount of downlink data in comparison with amount of uplink data where the performance of our low-complexity algorithm for the IP-CSI scenario is investigated. Specifically, we set the ratio between the amount of downlink data and the amount of uplink data (Γ_{dpu}) (i.e., computed as $\sum_{l_k \in \mathcal{L}_k} b^{\mathsf{dl}}_{k,l_k} / \sum_{l_k \in \mathcal{L}_k} b_{k,l_k}$) equal to 0.5, 1, and 1.5 corresponding to low downlink data ('$\Gamma_{\mathsf{dpu}} = 0.5$'), medium downlink data ('$\Gamma_{\mathsf{dpu}} = 1$'), and large downlink data ('$\Gamma_{\mathsf{dpu}} = 1.5$'), respectively.

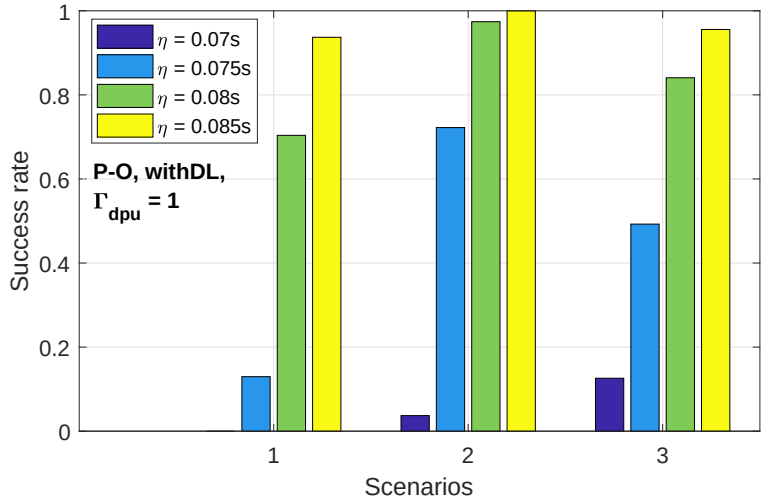

Figure 5.5 – Success rate for task processing with $\Gamma_{\mathsf{dpu}} = 1$.

It can be observed from this figure that the low-complexity algorithm achieves close-to-optimal performance when the maximum delay constraint is less stringent ('blue' and 'black' curves). For the IP-CSI scenario, mobile users will require more energy for data transmission to compensate for the CSI estimation errors. When the amount of downlink data becomes larger, more time is required to transfer the download data which means that less time is available for uploading the uplink data and computation at the cloud server. In some cases, increasing the transmit power to its maximum value may not lead to improved SINR, and the low transmission rate may prevent successful uplink data transmission in the offloading process. In all studied scenarios, even for the high value of Γ_{dpu}, the partial offloading scheme enables us to save energy significantly.

Fig. 5.5 shows the success rate for which computation task processing can be completed successfully in the MEC system with limited computing resource and three different scenarios: 1) $\sum_{l_k \in \mathcal{L}_k} b_{k,l_k} = 1$ Mbits, $F^{\mathsf{c}} = 40$ GHz; 2) $\sum_{l_k \in \mathcal{L}_k} b_{k,l_k} = 0.75$ Mbits, $F^{\mathsf{c}} = 40$ GHz, and 3) $\sum_{l_k \in \mathcal{L}_k} b_{k,l_k} = 1$ Mbits, $F^{\mathsf{c}} = 80$ GHz. The success rate is obtained by calculating the ratio between the number of successful computations and the total 200 different realizations. The results are obtained for the optimal algorithm, P-CSI and no downlink data transmission. We can see from this figure that, with the same requirement on computation, the larger amount of computing resource available at the cloud server and the smaller amount of uplink data, the better performance (i.e., lower latency and higher success rate) that mobile users can achieve through computation offloading.

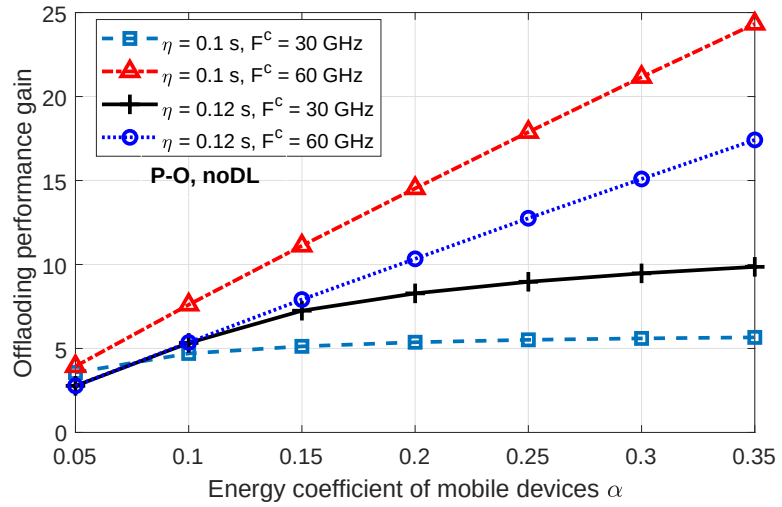

Figure 5.6 – Performance gain versus energy coefficient of mobile device.

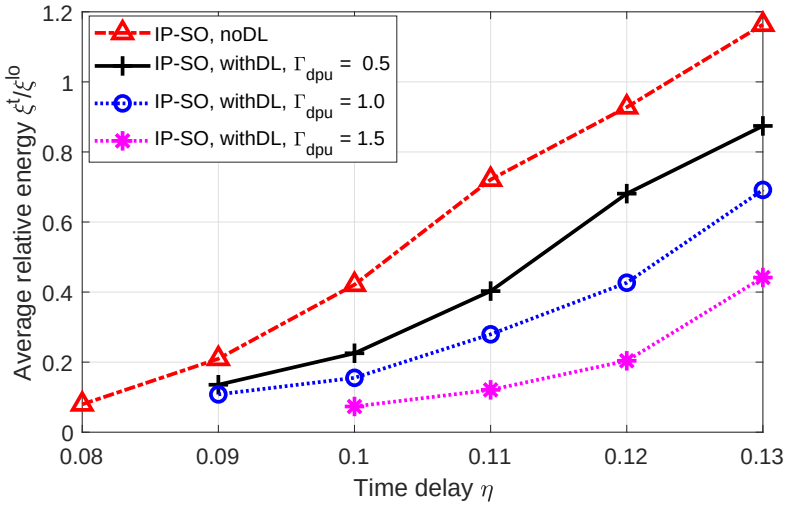

Figure 5.7 – Average ratio of energy components versus time delay.

Fig. 5.6 illustrates the offloading performance gain versus the energy coefficient of mobile devices. This performance gain is computed as $(\xi^{\mathsf{no-offload}} - \xi)/\xi$, which is used to compare the relative difference between the offloading and no-offloading cases. As shown in Fig. 5.6, larger performance gain can be obtained for applications with more stringent delay requirement. Moreover, the resource-rich cloud can lead to a significant performance gain for low-cost devices equipped with chipsets having higher coefficient α.

Fig. 5.7 presents the ratio between the energy components due to transmission and local processing $(\xi^{\mathsf{t}}/\xi^{\mathsf{lo}})$ for different values of Γ_{dpu}, where $\xi^{\mathsf{t}} = \frac{1}{K}\sum_k \xi_k^{\mathsf{t}}$ and $\xi^{\mathsf{lo}} = \frac{1}{K}\sum_k \xi_k^{\mathsf{lo}}$. It can be observed that more energy will be needed for transmitting data when the maximum allowable delay

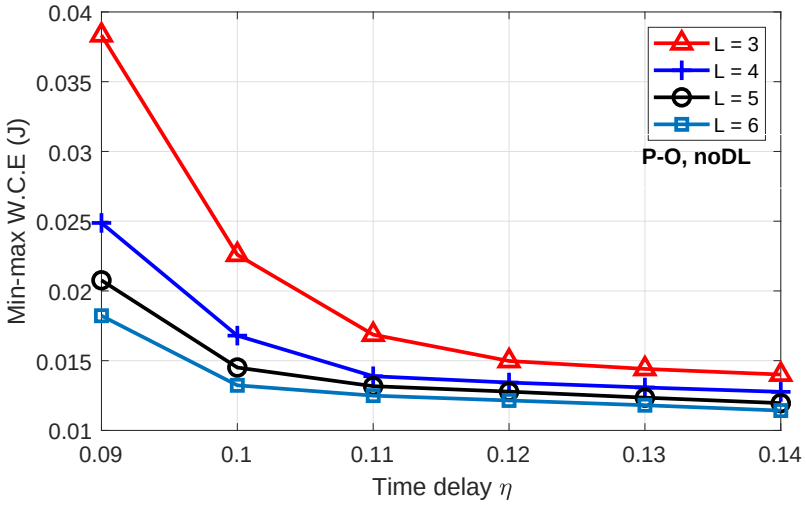

Figure 5.8 – Performance with difference number of parallel tasks.

is smaller; otherwise, more energy will be used for local computation. In fact, the previous figure suggests that if the system has enough resource to guarantee the successful data transmission and remote computation, computation offloading is preferred to save energy. Therefore, larger number of tasks can be processed remotely and less energy is used for executing tasks locally for larger maximum allowable delay. It means that the proportion of local computation energy will becomes smaller. Moreover, when the required transmit energy is smaller than ones for local computation, the increase in the transmit energy to compensate for the download time will become smaller than the increase in the energy required for computing more tasks, which cannot be offloaded with larger Γ_{dpu}. Thus, the relative energy ratio ξ_t/ξ^{lo} will also decrease.

The achieved W.C.E versus allowable delay is illustrated in Fig. 5.8 for different number of parallel tasks per UE L. It can be observed that the min-max W.C.E decreases quite drastically as the number of tasks increases, especially in the regime with a small number of parallel tasks. When L increases to a sufficiently large value for which the radio and computing resources can be effectively allocated to all UEs, the difference in performance due to increasing L will become insignificant.

Fig. 5.9 shows the fairness achieved for different UEs when applying the proposed min-max based computation offloading strategy in the IP-CSI scenario with no download data and $\eta = 0.1$s. On the average, each UE offloads more than half of its required computation demand and the resulting consumed energy is fairly similar among the UEs. Furthermore, the average weighted consumed energy for each user is quite smaller than average min-max W.C.E (shown in Fig. 5.4). This is

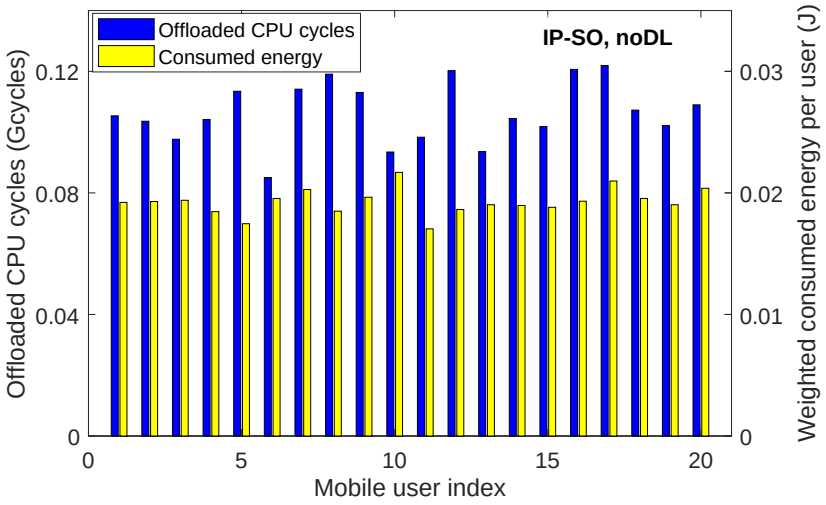

Figure 5.9 – Computation allocation and total consumed energy with allowable latency of 0.1s.

reasonable for the underlying min-max strategy because the min-max W.C.E value corresponds to the highest weighted consumed energy among all users in the system.

5.8 Conclusion

In this paper, we have developed both optimal and low-complexity algorithms to tackle general joint computation offloading and resource allocation for the MIMO based mobile cloud computing system considering P-CSI and IP-CSI. Our numerical studies have confirmed that significant energy saving of the proposed design compared to the no-offloading scenario can be achieved; the proposed sub-optimal algorithm achieves close to optimal performance when the delay constraint is not very stringent; and our proposed designs can provide great fairness for different users.

In the current work, we assume that different user's tasks are independent. We plan to address the offloading scenario for the MIMO-based MEC system where user's tasks are dependent in our future work. Furthermore, offloading design for the MIMO-based multi-task multi-user setting in the hierarchical fog-cloud system will be considered.

5.9 Appendices

5.9.1 Proof of Proposition 5.3

It can be verified that the transmission time in $(5.19c)_k$ decreases with p_k. Therefore, f_k^{c} achieves its minimum value when p_k is equal to its largest possible value. Moreover, f_k^{c} can be greater than zero if and only if the minimum transmission time is less than the maximum allowable delay, i.e., $\frac{b_k^{\mathsf{a}}}{W\log_2(1+p_k\beta_k^{\mathsf{a}})}<\eta_k$. If $g_k(p_k)$ is less than or equal to zero, all constraints are satisfied at p_k and the objective function achieves its minimum value. Otherwise, we will consider the case where $g_k(p_k)>0$. The constraints $(5.19c)_k$ and $(5.19g)_k$ can be rewritten as $p_k^{\mathsf{a}}<p_k<p_k$. On the other hand, it can be verified that $g_k(p_k)$ is a convex function of p_k. Therefore, from the Karush Kuhn Tucker (KKT) conditions which can be used to find the maximum value of p_k satisfying constraint $(5.26a)_k$ and $p_k^{\mathsf{a}}<p_k^\star<p_k$, we can deduce that the optimal $p_k^\star$ must satisfy $g_k(p_k^\star)=0$.

5.9.2 Proof of Proposition 5.5

For given values of ξ and f_k^{c}, constraints (5.20b) and (5.19c) can be rewritten as

$$\begin{aligned}\hat{r}_k^{\mathsf{lb}} &= W\log_2(p_k+\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k)-W\log_2(\boldsymbol{p}^{\mathsf{t}}\boldsymbol{\lambda}_k+\sigma_k)\\ &\geq \max\left(\left(\frac{T}{T-\tau}\right)\frac{b_k^{\mathsf{a}}}{\eta_k-\frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}}},\left(\frac{\tau(p^{\mathsf{tr}}+p_{k,0})}{T-\tau}+p_k+p_{k,0}\right)\frac{b_k^{\mathsf{a}}}{\frac{\xi}{w_k}-\xi_k^{\mathsf{lo}}}\right).\end{aligned}\tag{5.42}$$

The left-hand-side of (5.42) is the sum of concave and convex functions; hence, it is a sigmoidal function. Consequently, $(\mathcal{P}_2^{\mathsf{PA}})$ with given values of ξ and f_k^{c} is a sigmoidal program, which is NP-hard and NP-hard to approximate [121]. Thus, the original subproblem $(\mathcal{P}_2^{\mathsf{PA}})$ is also NP-hard.

5.9.3 Proof of Proposition 5.6

Let $g_{1,k}^{(q)}(\boldsymbol{p}) = p_k + p_{k,0} + \frac{\tau(p^{\mathsf{tr}}+p_{k,0})}{T-\tau} - \xi^{\mathsf{a}}\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)})$ and $g_{2,k}^{(q)}(\boldsymbol{p}, f_k^{\mathsf{c}}) = (\frac{T}{T-\tau})\frac{b_k^{\mathsf{a}}}{\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)})} + \frac{c_k^{\mathsf{a}}}{f_k^{\mathsf{c}}} - \eta_k$. We have

$$\begin{aligned}
\chi^{(q)} &= \max_k \max\{g_{1,k}^{(q)}(\boldsymbol{p}^{(q)}), g_{2,k}^{(q)}(\boldsymbol{p}^{(q)}, (f_k^{\mathsf{c}})^{(q)}\} \geq \min_{\boldsymbol{p}} \max_k \max\{g_{1,k}^{(q)}(\boldsymbol{p}), g_{2,k}^{(q)}(\boldsymbol{p}, (f_k^{\mathsf{c}})^{(q)})\} \\
&= \max_k \max\{g_{1,k}^{(q)}(\boldsymbol{p}^{(q+1)}), g_{2,k}^{(q)}(\boldsymbol{p}^{(q+1)}, (f_k^{\mathsf{c}})^{(q)})\} \overset{\text{(a)}}{\geq} \max_k \max\{g_{1,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}), g_{2,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}, (f_k^{\mathsf{c}})^{(q)})\} \\
&\geq \min_{f_k^{\mathsf{c}}} \max_k \max\{g_{1,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}), g_{2,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}, f_k^{\mathsf{c}})\} \\
&= \max_k \max\{g_{1,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}), g_{2,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}, (f_k^{\mathsf{c}})^{(q+1)})\} = \chi^{(q+1)},
\end{aligned}$$

where $\boldsymbol{p}^{(q+1)} = \operatorname{argmin}_{\boldsymbol{p}} \max_k \max\{g_{1,k}^{(q)}(\boldsymbol{p}), g_{2,k}^{(q)}(\boldsymbol{p}, (f_k^{\mathsf{c}})^{(q)})\}$, $(f_k^{\mathsf{c}})^{(q+1)} = \operatorname{argmin}_{f_k^{\mathsf{c}}} \max_k \max\{g_{1,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}), g_{2,k}^{(q+1)}(\boldsymbol{p}^{(q+1)}, f_k^{\mathsf{c}})\}$. Inequality (a) holds since

$$\tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}|\boldsymbol{p}^{(q)}) \leq \tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}^{(q+1)}|\boldsymbol{p}^{(q)}) \leq \hat{r}_k^{\mathsf{lb}}(\boldsymbol{p}^{(q+1)}) = \tilde{r}_k^{\mathsf{lb}}(\boldsymbol{p}^{(q+1)}|\boldsymbol{p}^{(q+1)}).$$

Therefore, for a given ξ, using D.C to approximate transmission rate creates a sequence of feasible and improving solutions for $(\mathcal{P}_{2,\mathsf{lP}}^{\mathsf{PA}})$, which, therefore, converges.

Chapter 6

Joint Data Compression and Computation Offloading in Hierarchical Fog-Cloud Systems

The content of this chapter was published in IEEE Transactions on Wireless Communications in the following paper:

Ti Ti Nguyen, Vu Nguyen Ha, Long B. Le, and Robert Schober "Joint Data Compression and Computation Offloading in Hierarchical Fog-Cloud Systems," *IEEE Trans. on Wireless Commun.,* vol. 19, no. 1, pp. 293–309, Jan. 2020.

Abstract

Data compression has the potential to significantly improve the computation offloading performance in hierarchical fog-cloud systems. However, it remains unknown how to optimally determine the compression ratio jointly with the computation offloading decisions and the resource allocation. This optimization problem is studied in this paper where we aim to minimize the maximum weighted energy and service delay cost (WEDC) of all users. First, we consider a scenario where data compression is performed only at the mobile users. We prove that the optimal offloading decisions have a threshold structure. Moreover, a novel three-step approach employing convexification techniques is developed to optimize the compression ratios and the resource allocation. Then, we address the more general design where data compression is performed at both the mobile users and the fog server. We propose three algorithms to overcome the strong coupling between the offloading decisions and the resource allocation. Numerical results show that the proposed optimal algorithm for data compression at only the mobile users can reduce the WEDC by up to 65% compared to computation offloading strategies that do not leverage data compression or use sub-optimal optimization approaches. The proposed algorithms with additional data compression at the fog server lead to a further reduction of the WEDC.

6.1 Introduction

Currently, mobile edge/cloud computing (MEC/MCC) technologies are considered as promising solutions for enhancing the mobile usability and prolonging the mobile battery life by offloading computation heavy applications to a remote fog/cloud server [7–9]. In an MCC system, enormous computing resources are available in the core network, but the limited backhaul capacity can induce significant delay for the underlying applications. In contrast, an MEC system, with computing resources deployed at the network edge in close proximity to the mobile devices, can enable computation offloading and meet demanding application requirements [10].

Hierarchical fog-cloud computing systems which leverage the advantages of both MCC and MEC can further enhance the system performance [11–15] where fog servers deployed at the network edge can operate collaboratively with the more powerful cloud servers to execute computation-intensive user applications. Specifically, when the users' applications require high computing power or low latency, their computation tasks can be offloaded and processed at the fog and/or remote cloud servers. However, the upsurge of mobile data and the constrained radio spectrum may result in significant delays in transferring offloaded data between the mobile users and the fog/cloud servers, which ultimately degrades the quality of service (QoS) [122]. To overcome this challenge, advanced data compression techniques can be leveraged to reduce the amount of incurred data (i.e., the input data of a user's application) [123, 124]. However, data compression entails additional computations needed for the execution of the corresponding compression and decompression algorithms [125]. Therefore, an efficient joint design of data compression, offloading decisions, and resource allocation is needed to take full advantage of data compression while meeting all QoS requirements and other system constraints.

6.1.1 Related Works

Computation offloading design for MCC/MCE systems has been studied extensively in the literature, see recent surveys [22, 33] and the references therein. Most existing works consider two main performance metrics for their designs, namely energy-efficiency [34–37] and delay-efficiency [38–41]. Focusing on energy-efficiency, the authors of [34] develop partial offloading frameworks for multiuser MEC systems employing time division multiple access (TDMA) and frequency-division multiple ac-

cess (FDMA). In [35], wireless power transfer is integrated into the computation offloading design. Moreover, different binary offloading frameworks are developed in [36, 37] where various branch-and-bound and heuristic algorithms are proposed to tackle the resulting mixed integer optimization problems.

Considering computation offloading from the delay-efficiency point of view, an iterative heuristic algorithm to optimize the binary offloading decisions for minimization of the overall computation and transmission delay in a hierarchical fog-cloud system is proposed in [38]. The authors in [39] formulate the computation offloading and resource allocation problem as a student-project-allocation game with the objective to maximize the ratio between the average offloaded data rate and the offloading cost at the users. In [40], the authors study a binary computation offloading problem for maximization of the weighted sum computation rate. Then, they propose a coordinate descent based algorithm in which the offloading decision and time-sharing variables are iteratively updated until convergence. Considering partial computation offloading, the authors in [41] propose a framework for minimization of the weighted-sum latency of the mobile users via collaborative cloud and fog computing assuming a TDMA based resource sharing strategy.

Some recently proposed schemes for computation offloading consider both energy and delay efficiency aspects [13, 15, 16]. In particular, the work in [13] proposes a radio and computing resource allocation framework where the computational loads of the fog and cloud servers are determined and the trade-off between power consumption and service delay is investigated. Additionally, the authors of [16] jointly optimize the transmit power and offloading probability for minimization of the average weighted energy, delay, and payment cost. In [15], the authors study fair computation offloading design minimizing the maximum weighted cost of delay and energy consumption among all users in a hierarchical fog-cloud system. In this work, a two-stage algorithm is proposed where the offloading decisions are determined in the first stage using a semidefinite relaxation and probability rounding based method while the radio and computing resource allocation is determined in the second stage. However, references [13, 15, 16, 34–40] have not exploited data compression for computation offloading.

There are few existing works that explore data compression for computation offloading. Specifically, the authors of [122] propose an analytical framework to evaluate the outage performance of a hierarchical fog-cloud system. Moreover, the work in [125] considers data compression for com-

putation offloading for systems with a single server but assumes a fixed compression ratio (i.e., this parameter is not optimized). In general, the compression ratio should be optimized jointly with the computation offloading decisions and the resource allocation to achieve optimal system performance. However, the computational load incurred by compression/decompression is a non-linear function of the compression ratio, which makes this joint optimization problem very challenging.

6.1.2 Contributions and Organization of the Paper

To the best of our knowledge, the joint design of data compression, computation offloading, and resource allocation for hierarchical fog-cloud systems has not been considered in the existing literature. The main contributions of this paper can be summarized as follows:

- We propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by data compression and decompression. In particular, the compression and decompression computational load as well as the quality of data recovery are modeled as functions of the compression ratio.
- For data compression at only the mobile users, we formulate the fair joint design of the compression ratio, computation offloading, and resource allocation as a mixed-integer non-linear programming (MINLP) optimization problem. This problem formulation takes into account practical constraints on the maximum transmit power, wireless access bandwidth, backhaul capacity, and computing resources. We propose an optimal algorithm, referred to as joint data compression, computation offloading, and resource allocation (JCORA) algorithm, which solves this challenging problem optimally. To develop this algorithm, we first prove that users incurring higher weighted energy and service delay cost (WEDC) when executing their application locally should have higher priority for offloading. Based on this result, the bisection search method is employed to optimally classify users into two user sets, namely the set of offloading users, and the set of remaining users, and JCORA globally optimizes the decision variables for both user sets.
- We then study a more general design where data compression is performed at both the mobile users and the fog server (with different compression ratios) before the compressed data are transmitted over the wireless link and the backhaul link to the fog server and the cloud server,

respectively. This enhanced design can lead to a significant performance gain when both the wireless access and the backhaul networks are congested. Three different solution approaches are proposed to solve this more general problem. In the first approach, we extend the design principle of the JCORA algorithm by employing the piece-wise linear approximation (PLA) method to tackle the coupling of the optimization variables. In the remaining approaches, we utilize the Lagrangian method and solve the dual optimization problem. Specifically, in the second approach, referred to as One-dimensional λ-Search based Two-Stage (OSTS) algorithm, a one-dimensional search is employed to determine the optimal value of the Lagrangian multiplier, while in the third approach, referred to as Iterative λ-Update based Two-Stage (IUTS) algorithm, a low-complexity iterative sub-gradient projection technique is adopted to tackle the problem.

- Extensive numerical results are presented to evaluate the performance gains of the proposed designs in comparison with conventional strategies that do not employ data compression. Moreover, our results confirm the excellent performance achievable by joint optimization of data compression, computation offloading decisions, and resource allocation in a hierarchical fog-cloud system.

The remainder of this paper is organized as follows. Section 6.2 presents the system model, the computation and transmission energy models, and the problem formulation. Section 6.3 develops the proposed optimal algorithm for the case when data compression is performed only at the mobile users. Section 6.4 provides the enhanced problem with data compression also at the fog server and three methods for solving it. Section 6.5 evaluates the performance of the proposed algorithms. Finally, Section 6.6 concludes this work.

6.2 System Model and Problem Formulation

6.2.1 System Model

We consider a hierarchical fog-cloud system consisting of K single-antenna mobile users, one cloud server, and one fog server co-located with a base station (BS) equipped with a large number of

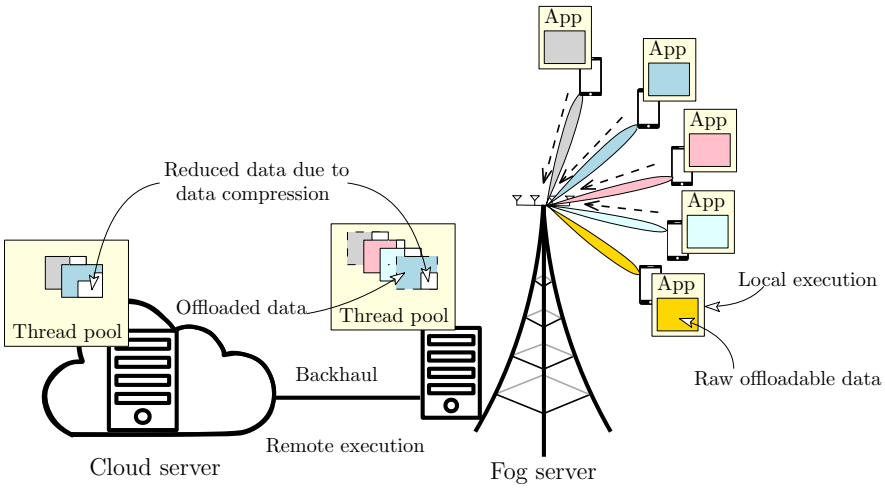

Figure 6.1 – Data compression and computation offloading in hierarchical fog-cloud systems.

antennas. In this system, the BS communicates with the users through wireless links while a (wired[1]) backhaul link is deployed between the BS co-located with the fog server and the cloud server as in Fig. 6.1. For convenience, we denote the set of users as $\mathcal{K}$. We assume that each user k needs to execute an application requiring c_k CPU cycles within an interval of $T_k^{\max}$ seconds, where $c_{k,0}$ CPU cycles must be executed locally at the mobile device and the remaining offloadable $c_{k,1}$ CPU cycles can be processed locally or offloaded and processed at the fog/cloud server for energy saving and delay improvement. Sequential processing of the unoffloadable and offloadable computing tasks is assumed in this paper. Let b_k^{in} be the number of bits representing the corresponding incurred data (i.e., programming states, input text/image/video) of the possibly-offloaded $c_{k,1}$ CPU cycles. To overcome the wireless transmission bottleneck caused by the capacity-limited wireless links between the users and the BS, data compression is employed at the users for reducing the amount of data transferred to the fog server.

In particular, once $c_{k,1}$ CPU cycles are offloaded, user k first compresses the corresponding b_k^{in} bits down to $b_k^{\mathsf{out,u}}$ bits before sending them to the remote fog server. The ratio between b_k^{in} and $b_k^{\mathsf{out,u}}$ is called the compression ratio and is denoted as $\omega_k^{\mathsf{u}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,u}}$. Depending on the available fog computing resources, the offloaded computation task can be directly processed at the fog server or be further offloaded to the cloud server. The amount of data required to represent the computation outcome sent back to the users is usually much smaller than that incurred by offloading the task. Therefore, similar to [15, 16, 34], we do not consider the downlink transmission of the computation results in this paper[2].

[1]The wireless backhaul will be considered in our future work.

[2]The design in this paper can be extended to also include the downlink transmission of feedback data as in [126].

Remark 6.1. *Running an application requires executing several unoffloadable sub-tasks that handle user interaction or access local I/O devices and cannot be executed remotely and other offloable sub-tasks that can be executed locally or remotely based on the employed offloading strategy [16, 26]. Practically, the workload corresponding to each sub-task of a specific application has to be pre-determined and remains unchanged according to the pre-programmed source code. Hence, the total workload of the offloadable components is typically fixed and cannot be optimized. In this work, we assume a binary offloading decision for all offloadable sub-tasks of each user. This corresponds to the practical scenario where all offloadable sub-tasks are strongly related such that they cannot be executed at different locations.*

6.2.1.1 Data Compression Model

Data compression can be achieved by eliminating only statistical redundancy (i.e., lossless compression) or by also removing unnecessary information (i.e., lossy compression). To realize it, compression and decompression algorithms must be executed at the data source and destination, respectively, which induces additional computational load. To the best of our knowledge, in the literature, there is no theoretical model for the computational workload incurred by data compression. Hence, we adopt a practical data-fitting approach to model the compression computational load, decompression computational load, and compression quality as non-linear functions of the compression ratio as follows:

$$c_k^{\mathsf{x,u}} = \gamma_{k,0}^{\mathsf{u}} \left[\gamma_{k,1}^{\mathsf{x,u}} (\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{x,u}}} + \gamma_{k,3}^{\mathsf{x,u}} \right], \text{ for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{6.1}$$

$$q_k^{\mathsf{qu,u}} = \gamma_{k,3}^{\mathsf{qu,u}} - \left[\gamma_{k,1}^{\mathsf{qu,u}} (\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{qu,u}}} \right], \text{ for } \omega_k^{\mathsf{u}} \in [\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}], \tag{6.2}$$

where 'x' = 'co' and 'de' stands for compression and decompression, respectively, $[\omega_{k,1}^{\mathsf{u,min}}, \omega_{k,1}^{\mathsf{u,max}}]$ represents the possible range of ω_k^{u} and depends on the compression algorithm employed at user k, $c_k^{\mathsf{co,u}}$ and $c_k^{\mathsf{de,u}}$ denote the additional CPU cycles at source and destination needed for compression and decompression, respectively[3]; $q_k^{\mathsf{qu,u}}$ represents the perceived QoS (i.e., this parameter, which is only considered for lossy compression, measures the deviation between the true data and the decompressed data); $\gamma_{k,0}^{\mathsf{u}}$ is the maximum number of CPU cycles; $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i = 1, 2, 3$, are constant

[3]Note that when the compression and decompression algorithms are executed at a fixed CPU clock speed, the computational load in CPU cycles is linearly proportional to the execution time.

parameters where $\gamma_{k,1}^{\mathsf{co/de/qu,u}}, \gamma_{k,3}^{\mathsf{co/de/qu,u}} \geq 0$. The values of the $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i = 1, 2, 3$, employed in this paper are determined based on experimental data collected by running the compression algorithms GZIP, BZ2, and JPEG in Python 3.0[4].

The accuracy of the proposed model is validated in Fig. 6.2 which illustrates the relation between the normalized compression/decompression execution time and the compression ratio using the lossless algorithms GZIP and BZ2 for the benchmark text files *"alice.txt"* and *"asyoulik.txt"* from Canterbury Corpus [127], and the lossy algorithm 'JPEG' for images *"clyde-river.jpg"* and *"frog.jpg"* from the Canadian Museum of Nature [128], obtained by simulating and fitting the proposed model. Here, the normalized execution time is the ratio of the actual execution time and the maximum execution time over all values of the compression ratio. The figure shows that the curves obtained through fitting using the proposed model match the simulation results well.

Remark 6.2. *A detailed comparison of the accuracy of the proposed compression computational load model and that of existing models is provided in Appendix G of our technical report [129].*

6.2.1.2 Computing and Offloading Model

We now introduce the binary offloading decision variables s_k^{u}, s_k^{f}, and s_k^{c} for the computation task of user k, where $s_k^{\mathsf{u}} = 1$, $s_k^{\mathsf{f}} = 1$, and $s_k^{\mathsf{c}} = 1$ denote the scenarios where the application is executed at the mobile device, the fog server, and the cloud server, respectively; and these variables are zero otherwise. Moreover, we consider binary offloading, thus the $c_{k,1}$ CPU cycles can be executed at exactly one location, which implies $s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 1$. Then, the total computational load of user k at the mobile device, denoted as c_k^{u}, and at the fog server, denoted as c_k^{f}, are given as, respectively,

$$c_k^{\mathsf{u}} = c_{k,0} + s_k^{\mathsf{u}} c_{k,1} + (1 - s_k^{\mathsf{u}}) c_k^{\mathsf{co,u}} \text{ and } c_k^{\mathsf{f}} = s_k^{\mathsf{f}} \left(c_{k,1} + c_k^{\mathsf{de,u}} \right). \tag{6.3}$$

[4]For validation, we collected three experimental data sets for three algorithms (GZIP, BZ2, or JPEG) by running each algorithm in Python 3.0 via a Linux terminal using Ubuntu 18.04.1 LTS on a computer equipped with CPU chipset Intel(R) core(TM) i7-4790 and 12 GB RAM. To keep the CPU clock speed almost constant, we turned off all other applications when executing the compression and decompression algorithms by using the *'cpupower tool'* in Linux. In each realization for each algorithm, we measured the execution time of running that algorithm with different compression ratios. Then, the experimental data sets are compiled from the average execution time for each compression ratio value over 1000 realizations of running each algorithm. This allowed us to estimate the normalized execution time, which is proportional to the normalized computational load.

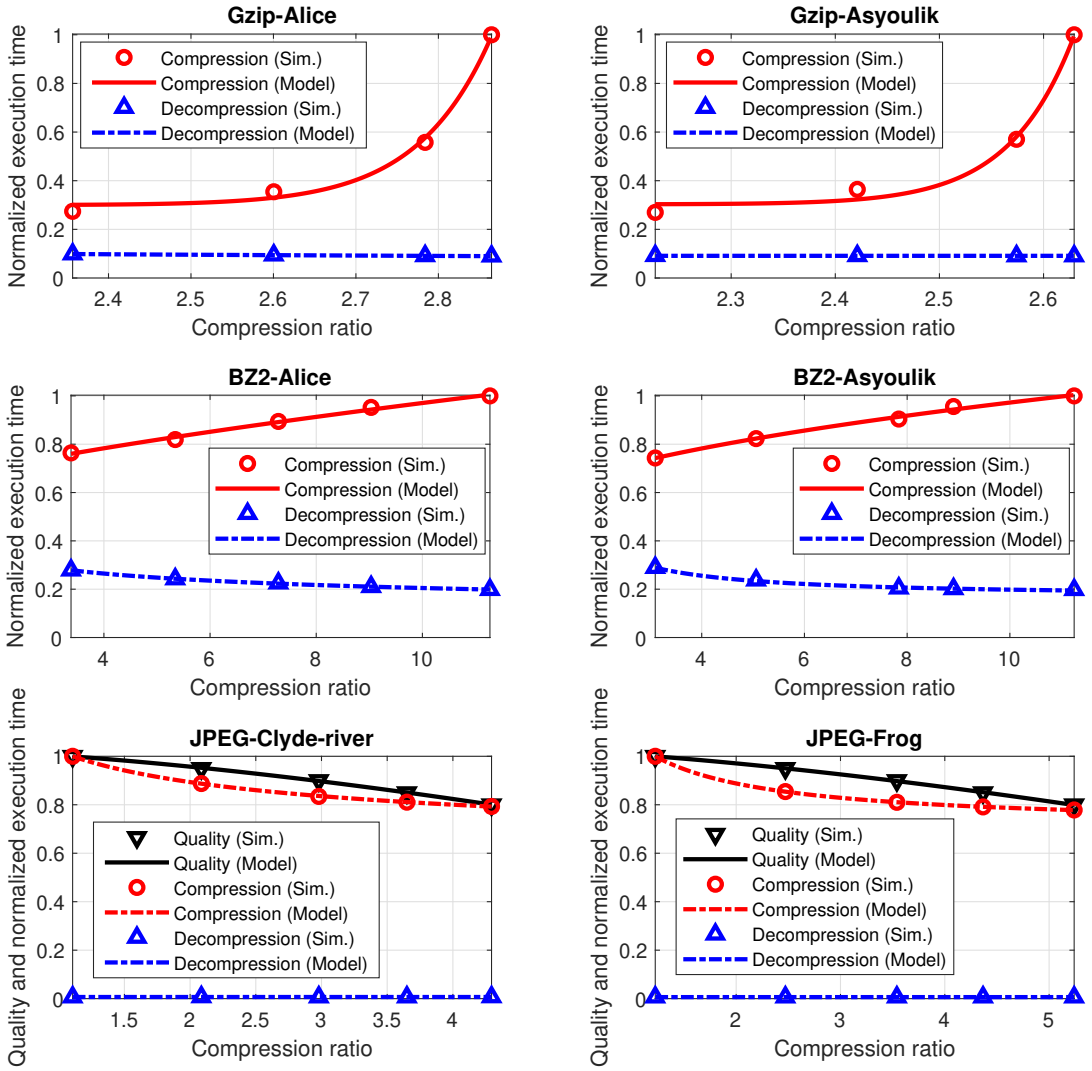

Figure 6.2 – Compression quality and normalized execution time.

As the fog and cloud servers are generally connected to the power grid while the capacity of a mobile battery is limited, we will focus on the energy consumption of the users [15]. The local computation energy consumed by user k and the local computation time can be expressed, respectively, as $\xi^{\mathsf{u}}_{1,k} = \alpha_k f^{\mathsf{u}2}_k c^{\mathsf{u}}_k$ and $t^{\mathsf{u}}_{1,k} = c^{\mathsf{u}}_k / f^{\mathsf{u}}_k$, where f^{u}_k is the CPU clock speed of user k and α_k denotes the energy coefficient specified by the CPU model [28]. Let f^{f}_k denote the CPU clock speed used at the fog server to process $c_{k,1}$. Then, the computing time at the fog server is given by $t^{\mathsf{f}}_{1,k} = c^{\mathsf{f}}_k / f^{\mathsf{f}}_k$. We assume that the computation task of each user is executed at the cloud server with a fixed delay of T^{c} seconds[5].

[5]The delay time for the cloud server consists of two components: the execution time and the CPU set-up time. Due to the huge computing resources at the cloud server, the execution time is generally much smaller than the CPU set-up time [130], which is identical for all users.

6.2.1.3 Communication Model

In order to send the incurred data during the offloading process, we assume that the channel state information is perfect and zero-forcing beamforming is applied at the BS and the average uplink rate from user k to the BS (fog server) is expressed as $r_k = \rho_k \log_2(1 + P_k \beta_{k,0})$, where P_k is the uplink transmit power per Hz of user k, ρ_k denotes the transmission bandwidth, and $\beta_{k,0} = M_0 \beta_k / \sigma_{\mathsf{bs}}$. Here, β_k represents the large-scale fading coefficient, σ_{bs} is the noise power density (watts per Hz), and M_0 is the multiple-input multiple-output (MIMO) beamforming gain, which is computed as $M - K$, where M is the number of antennas at the BS [25]. It is assumed that the number of antennas is sufficiently large so that M_0 is identical for all users. Then, the uplink transmission time and energy of user k can be computed, respectively, as $t^{\mathsf{u}}_{2,k} = (1 - s^{\mathsf{u}}_k) b^{\mathsf{out,u}}_k / r_k$ and $\xi^{\mathsf{u}}_{2,k} = \rho_k (P_k + P_{k,0}) t^{\mathsf{u}}_{2,k}$, where $P_{k,0}$ denotes the circuit power consumption per Hz. For the data transmission between the fog server and the cloud server, a backhaul link with capacity D^{max} bps (bits per second) is assumed. Let d_k denote the backhaul rate allocated to user k. Then, the transmission time from the fog server to the cloud server is $t^{\mathsf{f}}_{2,k} = s^{\mathsf{c}}_k b^{\mathsf{out,u}}_k / d_k$.

6.2.2 Problem Formulation

Assume the users have to pay for their usage of the radio and computing resources at the fog/cloud servers. Then, the service cost of user k can be modeled as $\Theta_k = (1 - s^{\mathsf{u}}_k)(w^{\mathsf{BW}} \rho_k + w^{\mathsf{C}} c_{k,1})$, where w^{BW} is the price per 1 Hz of bandwidth for wireless data transmission, and w^{C} is the price paid to execute one CPU cycle at the fog/cloud servers. Assuming that a pre-determined contract agreement specifies a maximum service cost Θ^{max}_k then $\Theta_k \leq \Theta^{\mathsf{max}}_k$. This constraint can be rewritten equivalently as $(1 - s^{\mathsf{u}}_k)\rho_k \leq \rho^{\mathsf{max}}_k = \frac{\Theta^{\mathsf{max}}_k - w^{\mathsf{C}} c_{k,1}}{w^{\mathsf{BW}}}$. Besides the constrained service cost, two important metrics for each user are the service latency and the consumed energy. Specifically, the total delay for completing the computation task of user k includes the computation delay of the mobile device, the average transmission delay of the mobile device, the computation delay of the fog server, the average transmission delay of the fog server over the backhaul link, and the computation delay of

the cloud server. Therefore, the total delay is given by

$$\begin{aligned} T_k &= t^{\mathsf{u}}_{1,k} + t^{\mathsf{u}}_{2,k} + t^{\mathsf{f}}_{1,k} + t^{\mathsf{f}}_{2,k} + s^{\mathsf{c}}_k T^{\mathsf{c}} \\ &= \frac{c_{k,0} + s^{\mathsf{u}}_k c_{k,1} + (1-s^{\mathsf{u}}_k) c^{\mathsf{co,u}}_k}{f^{\mathsf{u}}_k} + \frac{(1-s^{\mathsf{u}}_k) b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k \rho_k \log_2(1+P_k \beta_{k,0})} + \frac{s^{\mathsf{f}}_k \left(c_{k,1} + c^{\mathsf{de,u}}_k \right)}{f^{\mathsf{f}}_k} + \frac{s^{\mathsf{c}}_k b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k d_k} + s^{\mathsf{c}}_k T^{\mathsf{c}}. \end{aligned}$$

Since we assume massive MIMO transmission with zero-forcing beamforming, multiple mobile users can transmit their data to the fog server at the same time over the same frequency band. Unlike [41], we do not adopt the TDMA transmission protocol where the users are scheduled and have to wait for their turns to transmit their data in the uplink. For the considered massive MIMO system, time-based scheduling is not required since all users can transmit concurrently.

Furthermore, the overall energy consumed at user k for processing its task comprises the energy for local computation and for data transmission in the offloading case. Hence, the energy consumption of user k is given by

$$\xi_k = \xi^{\mathsf{u}}_{1,k} + \xi^{\mathsf{u}}_{2,k} = \alpha_k f^{\mathsf{u}2}_k (c_{k,0} + s^{\mathsf{u}}_k c_{k,1} + (1-s^{\mathsf{u}}_k) c^{\mathsf{co,u}}_k) + \frac{(P_k + P_{k,0})(1-s^{\mathsf{u}}_k) b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k \log_2(1+p_k \beta_{k,0})}. \tag{6.4}$$

Practically, all users want to save energy and enjoy low application execution latency. Hence, we adopt the WEDC as the objective function of each user k as follows:

$$\Xi_k = w^{\mathsf{T}}_k T_k + w^{\mathsf{E}}_k \xi_k,$$

where w^{T}_k and w^{E}_k represent the weights corresponding to the service latency and consumed energy, respectively. These weights can be pre-determined by the users to reflect their priorities or interests. The proposed design aims to minimize the WEDC function for each user while maintaining fairness

among all users. Towards this end, we consider the following min-max optimization problem:

$$(\mathcal{P}_1) \quad \min_{\Omega_1} \max_{k} \quad \Xi_k \tag{6.5a}$$

$$\text{s. t.} \quad f_k^{\mathsf{u}} \leq F_k^{\mathsf{max}}, \forall k, \tag{6.5b}$$

$$\sum_k f_k^{\mathsf{f}} \leq F^{\mathsf{f,max}}, \tag{6.5c}$$

$$s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}} \in \{0,1\}, \forall k, \tag{6.5d}$$

$$s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 1, \forall k, \tag{6.5e}$$

$$\omega_k^{\mathsf{u,min}} \leq \omega_k^{\mathsf{u}} \leq \omega_k^{\mathsf{u,max}}, \forall k, \tag{6.5f}$$

$$0 \leq \rho_k P_k \leq P_k^{\mathsf{max}}, \forall k, \tag{6.5g}$$

$$0 \leq \rho_k \leq \rho_k^{\mathsf{max}}, \ \forall k, \tag{6.5h}$$

$$\textstyle\sum_k d_k \leq D^{\mathsf{max}}, , \tag{6.5i}$$

$$T_k \leq T_k^{\mathsf{max}}, \forall k, \tag{6.5j}$$

where $\Omega_1 = \cup_{k\in\mathcal{K}}\Omega_{1,k}$, $\Omega_{1,k} = \{s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, \omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, P_k, \ \rho_k, d_k\}$; F_k^{max} is the maximum CPU clock speed of user k, $F^{\mathsf{f,max}}$ is the maximum CPU clock speed of the fog server, P_k^{max} is the maximum transmit power of user k, $[\omega_k^{\mathsf{u,min}}, \omega_k^{\mathsf{u,max}}]$ denotes the feasible range of the compression ratio ω_k^{u} which can guarantee the required QoS of the recovered data. In particular, for lossless data compression where the perceived QoS $q_k^{\mathsf{qu,u}} = 1$ for all ω_k^{u}, this feasible range is determined as $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ and $\omega_k^{\mathsf{u,max}} = \omega_{k,1}^{\mathsf{u,max}}$. For lossy data compression where the perceived QoS is required to be greater than $q_k^{\mathsf{qu,u,min}}$, this range is determined as $\omega_k^{\mathsf{u,min}} = \omega_{k,1}^{\mathsf{u,min}}$ and $\omega_k^{\mathsf{u,max}} = \min\left\{\omega_{k,1}^{\mathsf{u,max}}, \left((\gamma_{k,3}^{\mathsf{qu,u}} - q_k^{\mathsf{qu,u,min}})/\gamma_{k,1}^{\mathsf{qu,u}}\right)^{1/\gamma_{k,2}^{\mathsf{qu,u}}}\right\}$. In this problem, (6.5b) and (6.5c) represent the constraints on the computing resources at the users and at the fog server, respectively, while the offloading decision constraints are characterized by (6.5d) and (6.5e). The constraints on the compression ratio are captured by (6.5f), while (6.5g) and (6.5h) impose constraints on the maximum user transmit power and the bandwidth, respectively. Finally, (6.5i) and (6.5j) are the constraints due to the limited backhaul capacity[6] and delay, respectively.

[6]For practical scenarios, the development of sophisticated models for the communication delay over a shared backhaul link is a non-trivial task due to the complicated interactions between the routing algorithm and the other network functions (e.g. scheduling, buffering) [41]. This issue is outside the scope of this paper and left for future work. Similar to the existing work in [41], our current paper studies joint data compression and computation offloading in a hybrid fog-cloud computing system where we assume that a fixed backhaul communication capacity is allocated to each user. A fixed backhaul capacity allocation was also assumed in several recent works including [41, 47, 116].

6.3 Optimal Algorithm Design for Data Compression at only Mobile Users

6.3.1 Problem Transformation

To gain insight into its non-smooth min-max objective function, we recast $(\mathcal{P}_1)$ into the following equivalent problem:

$$
\begin{aligned}
(\mathcal{P}_2) \quad & \min_{\Omega_1 \cup \eta} \quad \eta && \text{(6.6a)} \\
& \text{s. t.} \quad \Xi_k \leq \eta, \forall k, && \text{(6.6b)} \\
& \qquad\quad (6.5b) - (6.5j),
\end{aligned}
$$

where η is an auxiliary variable. $(\mathcal{P}_2)$ is a MINLP problem which is difficult to solve due to the complex fractional and bilinear form of the transmission time and energy consumption, the logarithmic transmission rate function, and the mix of binary offloading decision variables and continuous variables. Conventional approaches usually decompose the problem into multiple sub-problems which optimize the offloading decision and the computing and radio resource allocation separately as in [15, 40] or relax the binary variables as in [36, 37]. These approaches can obtain only sub-optimal solutions.

To solve the problem optimally, we first study how to classify the users into two sets, namely, a *"locally executing user set"* which is the set of users executing their applications locally, and an *"offloading user set"* which is the set of users offloading their applications for processing at the fog/cloud server. This classification is important because, in all constraints of $(\mathcal{P}_2)$, the optimization variables corresponding to the locally executing users are independent from the optimization variables of the other users. Hence, the decisions for the locally executing users can be optimized by decomposing $(\mathcal{P}_2)$ into user independent sub-problems which can be solved separately. The optimal algorithm is developed based on the bisection search approach where in each search iteration, we perform: 1) user classification based on the current value of η using the results in Theorem 6.1 below; 2) feasibility verification for sub-problem $(\mathcal{P}_\mathcal{B})$ of $(\mathcal{P}_2)$ corresponding to the offloading user set $\mathcal{B}$; and 3) updates of lower and upper bounds on η according to the feasibility verification outcome. The detailed design is presented in the following.

Algorithm 6.1. Optimal Joint Data Compression, Offloading, and Resource Allocation (JCORA)

1: **Initialize**: Compute $\eta_k^{\mathsf{lo}}, \forall k \in \mathcal{K}$ as in (6.9), choose ϵ, assign $\eta^{\mathsf{min}} = 0$, $\eta^{\mathsf{max}} = \max_k(\eta_k^{\mathsf{lo}})$, and set BOOL = *False*.
2: **while** $(\eta^{\mathsf{max}} - \eta^{\mathsf{min}} > \epsilon)$ & (BOOL = *False*) **do**
3: Assign $\eta = (\eta_{\mathsf{max}} + \eta_{\mathsf{min}})/2$, and then define sets $\mathcal{A} = \{k|\eta_k^{\mathsf{lo}} \leq \eta\}$ and $\mathcal{B} = \mathcal{K}/\mathcal{A}$.
4: Check feasibility of $(\mathcal{P}_\mathcal{B})$ as in Section 6.3.3.
5: **if** $(\mathcal{P}_\mathcal{B})$ *is feasible* **then** $\eta^{\mathsf{max}} = \eta$, BOOL = *True*, **else** $\eta^{\mathsf{min}} = \eta$, BOOL = *False*, **end if**
6: **end while**

6.3.2 User Classification

Let $\mathcal{A}$ and $\mathcal{B}$ be the locally executing and the offloading user sets, respectively. We further define any pair of sets $(\mathcal{A}, \mathcal{B})$ satisfying $\mathcal{B} = \mathcal{K} \backslash \mathcal{A}$ as a user classification. By defining

$$\mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) = w_k^{\mathrm{E}} \alpha_k (f_k^{\mathsf{u}})^2 c_k + w_k^{\mathrm{T}} c_k / f_k^{\mathsf{u}}, \tag{6.7}$$

and $\Omega_\mathcal{B} = \cup_{k \in \mathcal{B}} \Omega_{1,k}$, then for a given classification $(\mathcal{A}, \mathcal{B})$, problem $(\mathcal{P}_2)$ can be tackled by solving two sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ for the users in sets $\mathcal{A}$ and $\mathcal{B}$, respectively, as follows:

$$\begin{aligned}
(\mathcal{P}_\mathcal{A}) \quad & \min_{\{f_k^{\mathsf{u}}\}_{k \in \mathcal{A}}, \eta} \quad \eta \\
& \text{s. t.} \quad (\text{CA0}): \ \mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) \leq \eta, \forall k \in \mathcal{A}, \\
& \qquad\quad (\text{CA2}): \ c_k / T_k^{\mathsf{max}} \leq f_k^{\mathsf{u}} \leq F_k^{\mathsf{max}}, \forall k \in \mathcal{A},
\end{aligned}$$

$$\begin{aligned}
(\mathcal{P}_\mathcal{B}) \quad & \min_{\Omega_\mathcal{B}, \eta} \quad \eta \\
& \text{s. t.} \quad (6.6b): \ \Xi_k \leq \eta, \forall k \in \mathcal{B}, \\
& \qquad\quad (6.5b) - (6.5j), \forall k \in \mathcal{B}.
\end{aligned}$$

Note that the variable set $\Omega_{1,k}$ corresponding to user k in $\mathcal{A}$ becomes $\{f_k^{\mathsf{u}}\}$ since we have $s_k^{\mathsf{u}} = 1$ and the other variables can be set equal to zero when user k executes its application locally. In such a scenario, Ξ_k can be simplified to $\mathcal{Q}_{k,0}(f_k^{\mathsf{u}})$. To attain more insight into the user classification, we now study the relationship between optimization sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ in the following lemma.

Lemma 6.1. [7] *We denote the optimal values of* $(\mathcal{P}_2)$, $(\mathcal{P}_\mathcal{A})$, *and* $(\mathcal{P}_\mathcal{B})$ *as* $\eta^\star$, $\eta^\star_\mathcal{A}$, *and* $\eta^\star_\mathcal{B}$, *respectively. Then, we have*

1. $\eta^\star \leq \max(\eta^\star_\mathcal{A}, \eta^\star_\mathcal{B})$ *for any classification* $(\mathcal{A}, \mathcal{B})$.
2. *The merged optimal solutions of* $(\mathcal{P}_\mathcal{A})$ *and* $(\mathcal{P}_\mathcal{B})$ *are the optimal solution of* $(\mathcal{P}_2)$ *if*

$$\eta^\star = \max(\eta^\star_\mathcal{A}, \eta^\star_\mathcal{B}). \tag{6.8}$$

3. *If* $\mathcal{B}' \subset \mathcal{B}$, *then we have* $\eta^\star_{\mathcal{B}'} \leq \eta^\star_\mathcal{B}$.

Considering Lemma 6.1, instead of solving $(\mathcal{P}_2)$, we can equivalently solve the two sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$. Moreover, a classification $(\mathcal{A}, \mathcal{B})$ is optimal if the condition in (6.8) holds. The optimal solution of $(\mathcal{P}_\mathcal{A})$ can be obtained as described in Proposition 6.1 while solving $(\mathcal{P}_\mathcal{B})$ requires a more complex approach which will be discussed in Section 6.3.4.

Proposition 6.1. [8] *The optimal objective value of* $(\mathcal{P}_\mathcal{A})$ *can be expressed as* $\eta^\star_\mathcal{A} = \max_{k\in\mathcal{A}} \eta^{\mathsf{lo}}_k$, *where* η^{lo}_k *is defined as*

$$\eta^{\mathsf{lo}}_k = \begin{cases} \mathcal{Q}_{k,0}(f^{\mathsf{u,sta}}_k), \ \textit{if } f^{\mathsf{u,sta}}_k \in [f^{\mathsf{u,min}}_k, F^{\mathsf{max}}_k] \\ \min\left(\mathcal{Q}_{k,0}(f^{\mathsf{u,min}}_k), \mathcal{Q}_{k,0}(F^{\mathsf{max}}_k)\right), \ \textit{otherwise}, \end{cases} \tag{6.9}$$

where $f^{\mathsf{u,min}}_k = c_k/T^{\mathsf{max}}_k$ *and* $f^{\mathsf{u,sta}}_k = \sqrt[3]{w^{\mathsf{T}}_k/(2w^{\mathsf{E}}_k\alpha_k)}$.

Based on the results in Lemma 6.1 and Proposition 6.1, the optimal user classification can be performed as described in the following theorem.

Theorem 6.1. *If* $\eta^\star$ *is the optimum objective value of problem* $(\mathcal{P}_2)$, *then an optimal classification,* $(\mathcal{A}^\star, \mathcal{B}^\star)$, *can be determined as* $\mathcal{A}^\star = \{k | \eta^{\mathsf{lo}}_k \leq \eta^\star\}$, *and* $\mathcal{B}^\star = \mathcal{K}\backslash\mathcal{A}^\star$.

Proof. The proof is given in Appendix 6.7.1. □

[7] Due to the space constraint, the proof of Lemma 6.1 is given in the online technical report [129].
[8] Due to the space constraint, the proof of Proposition 6.1 is given in the online technical report [129].

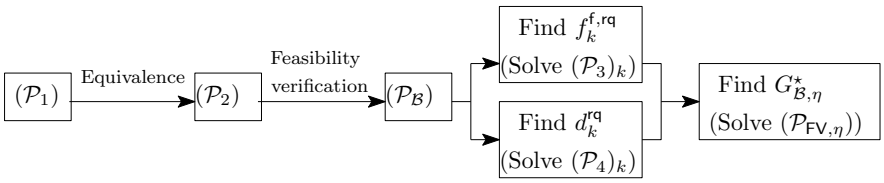

Figure 6.3 – Relationship between the (sub)problems when solving $(\mathcal{P}_1)$ by the JCORA algorithm.

6.3.3 General Optimal Algorithm Design

The results in Theorem 6.1 are now employed to develop an optimal algorithm for solving $(\mathcal{P}_2)$ by iteratively solving $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ and updating $(\mathcal{A}, \mathcal{B})$ until the optimal $(\mathcal{A}^\star, \mathcal{B}^\star)$ is obtained. The general optimal algorithm is presented in Algorithm 6.1. In this algorithm, we initially calculate η_k^{lo} for all users in $\mathcal{K}$ as in (6.9). Then, we employ the bisection search to find the optimum $\eta^\star$ where upper bound η^{max} and lower bound η^{min} are iteratively updated until the difference between them becomes sufficiently small, $(\mathcal{P}_\mathcal{B})$ is feasible, and the sets $\mathcal{A}$ and $\mathcal{B}$ do not change. At convergence, the optimal classification solution can be obtained by merging the solutions of $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$. The optimal solution of $(\mathcal{P}_\mathcal{A})$ can be determined using Proposition 6.1 and the verification of the feasibility of $(\mathcal{P}_\mathcal{B})$ is addressed in the following. The relationship between the (sub)problems when solving $(\mathcal{P}_1)$ is illustrated in Fig. 6.3.

6.3.4 Feasibility Verification of $(\mathcal{P}_\mathcal{B})$

In order to verify the feasibility of $(\mathcal{P}_\mathcal{B})$, we consider the following problem

$$(\mathcal{P}_{\mathsf{FV},\eta}) \quad \min_{\Omega_\mathcal{B}} \sum_{k\in\mathcal{B}} f_k^{\mathsf{f}} \quad \text{s. t.} \quad (6.6b), (6.5b), (6.5d) - (6.5j).$$

This problem minimizes the total required computing resource of the fog server subject to all constraints of $(\mathcal{P}_\mathcal{B})$ except $(6.5c)$. Let $G^\star_{\mathcal{B},\eta}$ be the objective value of problem $(\mathcal{P}_{\mathsf{FV},\eta})$. Then, the feasibility of $(\mathcal{P}_\mathcal{B})$ can be verified by comparing $G^\star_{\mathcal{B},\eta}$ to the available fog computing resource $F^{\mathsf{f,max}}$. In particular, problem $(\mathcal{P}_\mathcal{B})$ is feasible if $G^\star_{\mathcal{B},\eta} \leq F^{\mathsf{f,max}}$. Otherwise, $(\mathcal{P}_\mathcal{B})$ is infeasible.

We propose to solve $(\mathcal{P}_{\mathsf{FV},\eta})$ as follows. First, recall that there are two possible scenarios for executing the tasks of the users in set $\mathcal{B}$ (referred to as modes): *Mode 1* - task execution at the fog server, i.e., $s_k^{\mathsf{f}} = 1$; *Mode 2* - task execution at the cloud server, i.e., $s_k^{\mathsf{c}} = 1$. In addition, the fog computing resources are only required by the users in *Mode 1* and the backhaul resources are

only used by the users in *Mode 2*. Considering these two modes, a three-step solution approach is proposed to verify the feasibility of sub-problem $(\mathcal{P}_{\mathcal{B}})$ as follows. In Step 1, the minimum required fog computing resource of every user is determined by assuming that it is in *Mode 1*. This step is fulfilled by solving sub-problem $(\mathcal{P}_3)_k$ for every user k, see Section 6.3.4.1. In Step 2, the minimum required backhaul rate for each user is optimized by assuming that it is in *Mode 2*. This step can be accomplished by solving sub-problem $(\mathcal{P}_4)_k$ for every user k, see Section 6.3.4.2. In Step 3, using the results obtained in the two previous steps, problem $(\mathcal{P}_{\mathsf{FV},\eta})$ is equivalently transformed to a mode-mapping problem, see Section 6.3.4.3.

6.3.4.1 Step 1 - Minimum Fog Computing Resources for User $k \in \mathcal{B}$

If the application of user k is executed at the fog server, the minimum fog computing resource required for this application, denoted as $f_k^{\mathsf{f},\mathsf{rq}}$, can be optimized based on the following sub-problem:

$$(\mathcal{P}_3)_k \quad \min_{\Omega_{2,k}} \; f_k^{\mathsf{f}} \quad \text{s. t.} \quad s_k^{\mathsf{f}} = 1,\ (6.6b)_k,\ (6.5b)_k,\ (6.5f)_k - (6.5h)_k,\ (6.5j)_k,$$

where $\Omega_{2,k} = \{\omega_k^{\mathsf{u}}, f_k^{\mathsf{u}}, f_k^{\mathsf{f}}, P_k, \rho_k\}$, $(6.6b)_k$, $(6.5b)_k$, $(6.5f)_k - (6.5h)_k$, and $(6.5j)_k$ denote the respective constraints of user k corresponding to $(6.6b)$, $(6.5b)$, $(6.5f) - (6.5h)$, and $(6.5j)$. In sub-problem $(\mathcal{P}_3)_k$, the WEDC function Ξ_k consists of posynomials and other terms involving $\log(1+P_k\beta_{k,0})$. We can convert Ξ_k into a convex function via logarithmic transformation as follows. When $s_k^{\mathsf{f}} = 1$, all variables in set $\Omega_{2,k}$ must be positive to satisfy constraints (C0) and (6.5j); therefore, we can employ the following variable transformations: $\tilde{\omega}_k^{\mathsf{u}} = \log(\omega_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{u}} = \log(f_k^{\mathsf{u}})$, $\tilde{f}_k^{\mathsf{f}} = \log(f_k^{\mathsf{f}})$, $\tilde{P}_k = \log(P_k)$, and $\tilde{\rho}_k = \log(\rho_k)$. With these transformations, the objective function and all constraints of $(P_3)_k$ except $(6.6b)_k$ and $(6.5j)_k$ are converted into a linear form while the total delay and the WEDC in $(6.5j)_k$ and $(6.6b)_k$ can be rewritten, respectively, as

$$T_k = \frac{b_k^{\mathsf{in}} \mathrm{e}^{-\tilde{\omega}_k^{\mathsf{u}} - \tilde{\rho}_k}}{\log\left(1 + \beta_{k,0}\mathrm{e}^{\tilde{P}_k}\right)} + \mathcal{Q}_{k,1}, \tag{6.10}$$

$$\Xi_k = \frac{w_k^{\mathrm{E}} b_k^{\mathsf{in}} \left[\mathrm{e}^{\tilde{P}_k - \tilde{\omega}_k^{\mathsf{u}}} + p_{k,0}\mathrm{e}^{-\tilde{\omega}_k^{\mathsf{u}}}\right]}{\log\left(1 + \beta_{k,0}\mathrm{e}^{\tilde{P}_k}\right)} + w_k^{\mathrm{E}}\alpha_k\mathcal{Q}_{k,2} + w_k^{\mathrm{T}}T_k, \tag{6.11}$$

Algorithm 6.2. Feasibility Verification of $(\mathcal{P}_{\mathcal{B}})$

1: Solve $(\mathcal{P}_3)_k$ to find $f_k^{\mathsf{f},\mathsf{rq}}, \forall k \in \mathcal{B}$, as in Section 6.3.4.1.
2: Solve $(\mathcal{P}_4)_k$ to find $d_k^{\mathsf{rq}}, \forall k \in \mathcal{B}$, as in Section 6.3.4.2.
3: **if** $\exists k$ such that $s_k^{\mathsf{f}} + s_k^{\mathsf{c}} = 0$ **then**
4: Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible
5: **else**
6: Solve $(\mathcal{P}_{\mathsf{FV},\eta})$ to find $G^{\star}_{\mathcal{B},\eta}$, as in Section 6.3.4.3.
7: **if** $G^{\star}_{\mathcal{B},\eta} < F^{\mathsf{f},\mathsf{max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible **end if**
8: **end if**

where

$$\mathcal{Q}_{k,1} = \left(c_{k,0} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,3}^{\mathsf{co}}\right)\mathrm{e}^{-\tilde{f}_k^{\mathsf{u}}} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,1}^{\mathsf{co}}\mathrm{e}^{\left(-\tilde{f}_k^{\mathsf{u}}+\gamma_{k,2}^{\mathsf{co}}\tilde{\omega}_k^{\mathsf{u}}\right)} + \left(c_{k,1} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,3}^{\mathsf{de}}\right)\mathrm{e}^{-\tilde{f}_k^{\mathsf{f}}} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,1}^{\mathsf{de}}\mathrm{e}^{\left(-\tilde{f}_k^{\mathsf{f}}+\gamma_{k,2}^{\mathsf{de}}\tilde{\omega}_k^{\mathsf{u}}\right)}$$

$$\mathcal{Q}_{k,2} = \left(c_{k,0} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,3}^{\mathsf{co}}\right)\mathrm{e}^{2\tilde{f}_k^{\mathsf{u}}} + \gamma_{k,0}^{\mathsf{u}}\gamma_{k,1}^{\mathsf{co}}\mathrm{e}^{\left(2\tilde{f}_k^{\mathsf{u}}+\gamma_{k,2}^{\mathsf{co}}\tilde{\omega}_k^{\mathsf{u}}\right)}.$$

The convexity of $(P_3)_k$ is formally stated in the following proposition.

Proposition 6.2. *Sub-problem $(\mathcal{P}_3)_k$ is convex with respect to set $\tilde{\Omega}_{2,k} \cup \tilde{l}_k$, where $\tilde{l}_k = \tilde{\omega}_k^{\mathsf{u}} + \tilde{\rho}_k$ and $\tilde{\Omega}_{2,k} = \{\tilde{\omega}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{u}}, \tilde{f}_k^{\mathsf{f}}, \tilde{P}_k, \tilde{\rho}_k\}$.*

Proof. The proof is given in Appendix 6.7.2. □

Based on Proposition 6.2, we can apply the interior point method to find the optimal solution $\tilde{\Omega}^{\star}_{2,k} = \{\tilde{\omega}_k^{\mathsf{u}\star},\ \tilde{f}_k^{\mathsf{u}\star},\ \tilde{f}_k^{\mathsf{f}\star},\ \tilde{p}_k^{\star},\ \tilde{\rho}_k^{\star}\}$ of $(\mathcal{P}_3)_k$ [1]. The optimal solution $\Omega^{\star}_{2,k} = \{\omega_k^{\mathsf{u}\star},\ f_k^{\mathsf{u}\star},\ f_k^{\mathsf{f}\star},\ P_k^{\star},\ \rho_k^{\star}\}$ can then be obtained from $\tilde{\Omega}^{\star}_{2,k}$. If $(\mathcal{P}_3)_k$ is infeasible, we set $s_k^{\mathsf{f}} = 0$. It is noted that $f_k^{\mathsf{f}\star}$ is also the value of $f_k^{\mathsf{f},\mathsf{rq}}$.

6.3.4.2 Step 2 - Minimum Allocated Backhaul Resource for User $k \in \mathcal{B}$

If the application of user k is executed at the cloud server, the minimum backhaul capacity for transferring its application to the cloud server, denoted as d_k^{rq}, can be determined by solving the following sub-problem:

$$(\mathcal{P}_4)_k \quad \min_{\Omega_{2,k} \cup d_k \setminus f_k^{\mathsf{f}}} \ d_k \quad \text{s. t.} \quad s_k^{\mathsf{c}} = 1, (6.6b)_k,\ (6.5b)_k,\ (6.5f)_k\text{–}(6.5h)_k,\ (6.5j)_k.$$

Similar to $(\mathcal{P}_3)_k$, $(\mathcal{P}_4)_k$ can be converted to a convex problem via logarithmic transformation; thus, we can find the optimal point d_k^{rq}. If $(\mathcal{P}_4)_k$ is infeasible, we set $s_k^{\mathsf{c}} = 0$.

6.3.4.3 Step 3 - Feasibility Verification

With the obtained values $f_k^{\mathsf{f,rq}}$ and d_k^{rq}, problem $(\mathcal{P}_{\mathsf{FV},\eta})$ can be transformed to

$$\begin{aligned}(\mathcal{P}_{\mathsf{FV},\eta}) \quad & \min_{\Omega_3} \ \mathcal{G}_{\mathcal{B},\eta}(\Omega_3) = \sum_{k\in\mathcal{B}}(1 - s_k^{\mathsf{c}}) f_k^{\mathsf{f,rq}} \\ & \text{s. t.} \ \sum_{k\in\mathcal{B}} s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \leq D^{\mathsf{max}}, \ s_k^{\mathsf{c}} \in \{0,1\},\end{aligned}$$

where $\Omega_3 = \{s_k^{\mathsf{c}} | k \in \mathcal{B}\}$ for a given η. In fact, $(\mathcal{P}_{\mathsf{FV},\eta})$ is a *"0-1 knapsack"* problem [29], which can be solved optimally and effectively using the CVX solver. If $G^{\star}_{\mathcal{B},\eta} \leq F^{\mathsf{f,max}}$, combining the set of all solutions of the $(\mathcal{P}_3)_k$'s, $(\mathcal{P}_4)_k$'s, and $(\mathcal{P}_{\mathsf{FV},\eta})$ yields a feasible solution of $(\mathcal{P}_{\mathcal{B}})$ for this value of η. Hence, $(\mathcal{P}_{\mathcal{B}})$ is feasible in such scenario. The feasibility verification of $(\mathcal{P}_{\mathcal{B}})$ is summarized in Algorithm 6.2.

6.3.5 Optimal JCORA Algorithm to Solve $(\mathcal{P}_2)$

Based on the results presented in the previous sections, the solution of $(\mathcal{P}_2)$ can be found by employing Algorithm 6.1 and the $(\mathcal{P}_{\mathcal{B}})$ feasibility verification presented in Algorithm 6.2. The optimality of the obtained solution is formally stated in the following theorem.

Theorem 6.2. *The integration of Algorithm 6.2 into Algorithm 6.1 yields the global optimum of MINLP* $(\mathcal{P}_2)$.

Proof. Algorithm 6.2 verifies the feasibility of $(\mathcal{P}_{\mathcal{B}})$ for any given value of $\eta_{\mathcal{B}} = \eta$. Therefore, if Algorithm 6.1 employs Algorithm 6.2, $(\mathcal{P}_2)$ is solved optimally. Note that after convergence, the optimal variables are given by the optimal solution of $(\mathcal{P}_3)_k$ if $s_k^{\mathsf{f}} = 1$ or $(\mathcal{P}_4)_k$ if $s_k^{\mathsf{c}} = 1$ where the values of the s_k^{f}'s and s_k^{c}'s are the outcomes of $(\mathcal{P}_{\mathsf{FV},\eta})$. □

6.3.6 Complexity Analysis

We analyze the computational complexity of the JCORA algorithm (Algorithm 6.2 is integrated into Algorithm 6.1) in terms of the required number of arithmetic operations. In Algorithm 6.1, the while-loop for the bisection search of η requires $\log_2(\frac{\eta^{\max}-\eta^{\min}}{\epsilon})$ iterations. To verify the feasibility of $(\mathcal{P}_{\mathcal{B}})$ for a given η, the convex problems $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ can be solved by using the interior point method with complexity $\mathcal{O}(m_1^{1/2}(m_1+m_2)m_2^2)$, where m_1 is the number of equality constraints, m_2 represents the number of variables [131], and $\mathcal{O}$ denotes the big-O notation. It can be verified that $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ have the same complexity. On the other hand, the knapsack problem $(\mathcal{P}_{\mathsf{FV},\eta})$ for $|\mathcal{B}|$ users can be solved by Algorithm 6.2 in pseudo-polynomial time with complexity $\mathcal{O}(\nu_1|\mathcal{B}|)$, where ν_1 is determined by the coefficients in $(\mathcal{P}_{\mathsf{FV},\eta})$ [29]. Moreover, $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ can be solved independently for all users $k \in \mathcal{B}$; therefore, the complexity of each bisection search step can be expressed as $|\mathcal{B}|\mathcal{O}((\mathcal{P}_3)_k)+|\mathcal{B}|\mathcal{O}((\mathcal{P}_4)_k)+\mathcal{O}(\mathcal{P}_{\mathsf{FV},\eta}) = \mathcal{O}(\nu_2|\mathcal{B}|)$, where $\nu_2 = \nu_1+2m_1^{1/2}(m_1+m_2)m_2^2$. Consequently, the overall complexity of the JCORA algorithm is $\mathcal{O}(\log_2(\frac{\eta^{\max}-\eta^{\min}}{\epsilon})\nu_2 K)$, i.e., $|\mathcal{B}| \leq K$.

6.4 Data compression at Both Mobile Users and Fog Server

We now consider the more general case where the fog server also performs data compression before transmitting the compressed data over the backhaul link to the cloud server. This design option can further enhance the performance for systems with a congested backhaul link. The backhaul compression ratio is defined as $\omega_k^{\mathsf{f}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,f}}$ where $b_k^{\mathsf{out,f}}$ stands for the number of bits transmitted over the backhaul link. Note that if $b_k^{\mathsf{out,f}} = b_k^{\mathsf{out,u}}$, then no data compression is employed at the fog server, which corresponds to the design in Section 6.3. Hence, *Mode 2* in Section 6.3.4.1 is equivalent to the scenario that the task is executed at the cloud server without data compression at the fog server. However, the fog server can re-compress the data before transmitting it to the cloud server for processing, which is referred to as *Mode 3* in the following. Denote s_k^{m} as the binary variable indicating whether or not data compression is performed at the fog server for user k ($s_k^{\mathsf{m}} = 1$ for data compression, and $s_k^{\mathsf{m}} = 0$, otherwise). Then, we have $s_k^{\mathsf{f}} = 1$ if user k is in *Mode 1*; $s_k^{\mathsf{c}} = 1$ if user k is in *Mode 2*; $s_k^{\mathsf{m}} = 1$ if user k is in *Mode 3*. In this general case, constraints

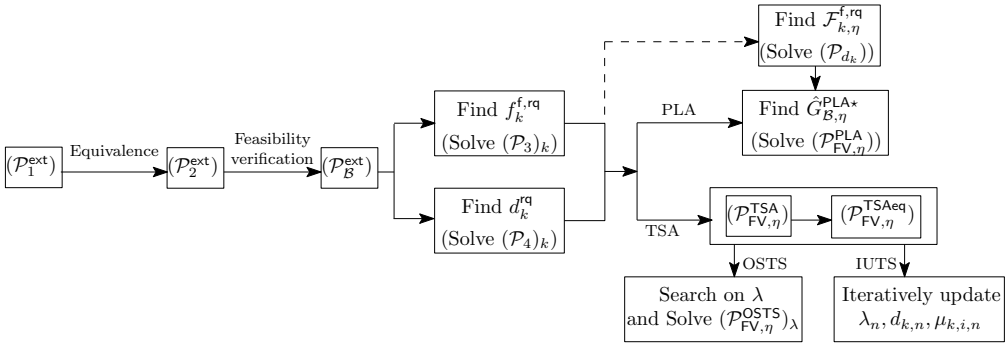

Figure 6.4 – Relationship between the (sub)problems when solving $(\mathcal{P}_1^{\mathsf{ext}})$.

(6.5d) and (6.5e) can be rewritten as

$$s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}} \in \{0,1\}, \forall k \in \mathcal{K}, \tag{6.12a}$$

$$s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + s_k^{\mathsf{m}} = 1, \forall k \in \mathcal{K}, \tag{6.12b}$$

Then, the computational load for compression and the output data corresponding to *Mode 3* can be modeled as $c_k^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}}\left[\gamma_{k,1}^{\mathsf{co,f}}(\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \gamma_{k,3}^{\mathsf{co,f}}\right]$ and $b_k^{\mathsf{out,f}} = b_k^{\mathsf{in}}/\omega_k^{\mathsf{f}}$, respectively, where $\gamma_{k,0}^{\mathsf{f}}, \gamma_{k,1}^{\mathsf{co,f}}, \gamma_{k,3}^{\mathsf{co,f}} \in \mathbb{R}_+$ are positive numbers. Here, we have additional constraints for the compression ratio at the fog server as

$$\omega_k^{\mathsf{f}} \in [\omega_k^{\mathsf{f,min}}, \omega_k^{\mathsf{f,max}}],\ \forall k \in \mathcal{K}. \tag{6.13}$$

Then, the total computational load for user k at the fog server becomes $\check{c}_k^{\mathsf{f}} = s_k^{\mathsf{f}}\left(c_{k,1} + c_k^{\mathsf{de,u}}\right) + s_k^{\mathsf{m}}(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})$, and the computing time at the fog server is $\check{t}_{1,k}^{\mathsf{f}} = \check{c}_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. Moreover, the transmission time incurred by offloading the data of user k from the fog server to the cloud server can be rewritten as $\check{t}_{2,k}^{\mathsf{f}} = \left(s_k^{\mathsf{f}} b_k^{\mathsf{out,u}} + s_k^{\mathsf{m}} b_k^{\mathsf{out,f}}\right)/d_k$. Then, the total delay for completing the computation task of user k is given by $\check{T}_k = t_{1,k}^{\mathsf{u}} + t_{2,k}^{\mathsf{u}} + \check{t}_{1,k}^{\mathsf{f}} + \check{t}_{2,k}^{\mathsf{f}} + (s_k^{\mathsf{c}} + s_k^{\mathsf{m}})T^{\mathsf{c}}$, and the WEDC becomes $\check{\Xi}_k = w_k^{\mathsf{T}}\check{T}_k + w_k^{\mathsf{E}}\xi_k$. Then, constraint (6.5j) is rewritten as

$$\check{T}_k \leq T_k^{\mathsf{max}}. \tag{6.14}$$

With the additional variables s_k^{m} and $\omega_k^{\mathsf{f}}, \forall k \in \mathcal{B}$, the extended versions of problems $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ can be stated, respectively, as

$$(\mathcal{P}_1^{\mathsf{ext}}) \quad \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\}} \max_k \ \check{\Xi}_k$$

$$\text{s. t.} \quad (6.5b), (6.5c), (6.5f)\text{–}(6.5i), (6.12a), (6.12b), (6.13), (6.14).$$

$$(\mathcal{P}_2^{\mathsf{ext}}) \quad \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\} \cup \eta} \quad \eta$$

$$\text{s. t.} \quad \check{\Xi}_k \leq \eta, \tag{6.16a}$$

$$(6.5b), (6.5c), (6.5f)-(6.5i), (6.12a), (6.12b), (6.13), (6.14).$$

The main challenge for solving the extended problem in comparison to the original one comes from the users in *Mode 3*. These users require both fog computing and backhaul resources. To solve the extended problem, we employ the general solution approach presented in Section 6.3 but modify the feasibility verification for $(\mathcal{P}_{\mathcal{B}})$. In particular, Algorithm 6.1 is used to determine sets $\mathcal{A}$ and $\mathcal{B}$ for a given η and we update η using the bisection search method. The results in Theorem 6.1 are still applicable for the extended problem. In the following, we propose several techniques for dealing with *Mode 3* and verify the feasibility of user classification for a given η in *Step 4* of Algorithm 6.1.

For a given η, $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is obtained by adding (6.13) to $(\mathcal{P}_{\mathcal{B}})$ and replacing Ξ_k and T_k by $\check{\Xi}_k$ and $\check{T}_k$, respectively. To verify the feasibility of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$, a similar three-step solution approach as for $(\mathcal{P}_{\mathcal{B}})$ is employed. In Steps 1 and 2, $f_k^{\mathsf{f,rq}}$ and d_k^{rq} which correspond to the users in *Mode 1* and *2* are optimized by solving $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ as in Sections 6.3.4.1 and 6.3.4.2, respectively. In Step 3, we first investigate the network resources required by the users in *Mode 3*, modify problem $(\mathcal{P}_{\mathsf{FV},\eta})$ to adapt it to the extended problem, and solve that problem to verify the feasibility. Three different methods for this extended problem will be proposed as follows.

In the first approach, we represent $f_k^{\mathsf{f,rq}}$ of user k in *Mode 3* as a function of d_k by employing a piece-wise linear approximation (PLA) method. Based on this approximation, we transform $(\mathcal{P}_{\mathsf{FV},\eta})$ into a standard mixed-integer linear programming (MILP) problem, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$, which can be solved effectively by using the CVX solver. In the other two approaches, we directly deal with the modified problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ without approximating $f_k^{\mathsf{f,rq}}$ of user k in *Mode 3*. To cope with this challenging MINLP problem, we first reduce the optimization variable set by exploiting some useful relations among the variables. Then, two algorithms are proposed to solve the resulting problem for the remaining variables. One algorithm is based on a one-dimensional search for the Lagrangian multiplier, see Section 6.4.2.1, while the other algorithm iteratively updates the

Algorithm 6.3. PLA-based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **Initialize**: L, η
2: Compute $f_k^{\mathsf{f},\mathsf{rq}}$ and d_k^{rq} for all $k \in \mathcal{B}$ as in Step 1 and 2 of Algorithm 6.2.
3: Define $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L,\ \forall k \in \mathcal{B}, l = 0 : L$.
4: Compute $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l})$. **If** $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l})$ is unbounded **then** Remove point $d_{k,l}$ **end if**.
5: Compute $A_{k,l}$, $B_{k,l}$, and then solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$ to get optimal value $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star}$ of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$.
6: **if** $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star} \leq F^{\mathsf{f},\mathsf{max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible **end if**

Lagrangian multiplier, see Section 6.4.2.2. The relationship between the (sub)problems when solving $(\mathcal{P}_1^{\mathsf{ext}})$ is illustrated in Fig. 6.4.

6.4.1 Piece-wise Linear Approximation based Algorithm (PLA)

After determining the minimum computing and backhaul resources, $f_k^{\mathsf{f},\mathsf{rq}}$ and d_k^{rq}, required in *Modes 1* and *2*, respectively, one can set $d_k \in (0, d_k^{\mathsf{rq}})$ for the users in *Mode 3*. We now study the relationship between f_k^{f} and d_k in *Mode 3* where user k demands both fog computing resources for re-compression and backhaul capacity resources. Towards this end, we determine the required fog computing resources for a given $d_k \in (0, d_k^{\mathsf{rq}})$ by solving the following problem:

$$(\mathcal{P}_{d_k}) \quad \min_{\Omega_{2,k} \cup \{\omega_k^{\mathsf{f}}\}} f_k^{\mathsf{f}}$$
$$\text{s. t. } s_k^{\mathsf{m}} = 1,\ (6.16a)_k,\ (6.5b)_k,\ (6.5f)_k - (6.5h)_k,\ (6.14)_k,\ (6.13)_k.$$

Let $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_k)$ be the optimal solution of this problem, which can be obtained by employing the logarithmic transformations described in Section 6.3.4.1. However, finding a closed-form expression for $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_k)$ is not tractable.

Hence, we propose to employ the *"Piece-wise Linear Approximation"* (PLA) method to divide the original domain into multiple small segments such that $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_k)$ can be approximated by a linear function in each segment. Suppose that the interval $[\epsilon_{\mathsf{d}}, d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}}]$ is divided into L segments of equal size, where ϵ_{d} is a very small number compared to d_k^{rq}, e.g, $\epsilon_{\mathsf{d}} = 1$. Specifically, the l^{th} segment corresponds to interval $[d_{k,l}, d_{k,l+1}]$, where $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L$ is a point such that $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l})$ and the value of the approximated function at this point are equal. Then, we can approximate $\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_k)$ as $\hat{\mathcal{F}}_{k,\eta}^{\mathsf{f},\mathsf{rq}}\Big(V_k, U_k\Big) = \sum_{l=0}^{L-1} \left(v_{k,l} A_{k,l} + u_{k,l} B_{k,l}\right)$, where $A_{k,l} = \Big(\mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l+1}) - \mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l})\Big)/(d_{k,l+1} - d_{k,l})$, $B_{k,l} = \mathcal{F}_{k,\eta}^{\mathsf{f},\mathsf{rq}}(d_{k,l}) - A_{k,l} d_{k,l}$, $V_k = \{v_{k,l}, l = 0, 1, ..., L-1\}$, $U_k = \{u_{k,l}, l = 0, 1, ..., L-1\}$, and contin-

uous variable $v_{k,l}$ and binary variable $u_{k,l}$ satisfy the following constraints:

$$s_k^{\mathsf{m}} = \sum_{l=0}^{L-1} u_{k,l} \leq 1, \forall k \in \mathcal{B}, \tag{6.17}$$

$$u_{k,l} d_{k,l} \leq v_{k,l} \leq u_{k,l+1} d_{k,l+1}, \forall k \in \mathcal{B}, l = 0, 1, ..., L-1. \tag{6.18}$$

Then, the allocated backhaul resources due to user k in *Mode 3* are rewritten as $s_k^{\mathsf{m}} d_k = \sum_{l=0}^{L-1} v_{k,l}$. Therefore, problem $(\mathcal{P}_{\mathsf{FV},\eta})$, which is used to determine the minimum total required fog computing resources for all users, is modified in this extended case as follows:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}}) \quad \min_{\check{\Omega}_3} \quad \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{PLA}}\left(\check{\Omega}_3\right) = \sum_{k\in\mathcal{B}} \left(s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} + \hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}\left(V_k, U_k\right)\right)$$

$$\text{s. t.} \quad s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, u_{k,l} \in \{0,1\}, \ \forall k, l, \tag{6.19a}$$

$$s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + \sum\nolimits_{l=0}^{L-1} u_{k,l} = 1, \tag{6.19b}$$

$$u_{k,l} d_{k,l} \leq v_{k,l} \leq u_{k,l+1} d_{k,l+1}, \ \forall k, l, \tag{6.19c}$$

$$\sum\nolimits_{k\in\mathcal{B}} \left(\sum\nolimits_{l=0}^{L-1} v_{k,l} + s_k^{\mathsf{c}} d_k^{\mathsf{rq}} \right) \leq D^{\mathsf{max}}, \tag{6.19d}$$

where $\check{\Omega}_3 = \cup_{k\in\mathcal{B}} \left(s_k^{\mathsf{f}} \cup s_k^{\mathsf{c}} \cup U_k \cup V_k \right)$ and constraints (6.19a), (6.19b), and (6.19c)-(6.19d) are the transformed constraints of original constraints (6.12a), (6.12b), and (6.5i), respectively. This transformed problem is an MILP problem, which can be solved effectively by using the CVX solver. The PLA based algorithm for verifying the feasibility of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is summarized in Algorithm 6.3, which can be integrated into Algorithm 6.1 to solve $(\mathcal{P}_2^{\mathsf{ext}})$. It is noted that if the value of $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ is unbounded for a given $d_{k,l}$, this infeasible point is removed when applying the PLA based algorithm.

6.4.2 Two-stage Solution Approach (TSA)

In this section, two two-stage algorithms are developed by exploiting the fact that the decompression computational load (and therefore, the associated energy consumption) is almost independent from the compression ratio as can be seen in Fig. 6.2. This implies that for a given η, the optimal values f_k^{u}, ω_k^{u}, P_k, and ρ_k for mobile user k are similar for both $s_k^{\mathsf{f}} = 1$ and $s_k^{\mathsf{c}} = 1$. Hence, in the first stage, after solving $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$, $\forall k \in \mathcal{B}$, introduced in Section 6.3, we can set these variables to the corresponding optimal solution of $(\mathcal{P}_3)_k$, denoted as $f_{k,1}^{\mathsf{u}\star}$, $\omega_{k,1}^{\mathsf{u}\star}$, $p_{k,1}^{\star}$, and $\rho_{k,1}^{\star}$. In the second stage, we find the remaining variables pertaining to the fog server $\Omega_4 = \cup_{k\in\mathcal{B}} \{s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}}, d_k, f_k^{\mathsf{f}}, \omega_k^{\mathsf{f}}\}$

by solving the following problem[9]:

$$(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta}) \quad \min_{\Omega_4} \quad \hat{\mathcal{G}}^{\mathsf{TSA}}_{\mathcal{B},\eta}(\Omega_4) = \sum_{k\in\mathcal{B}} \left(s^{\mathsf{m}}_k f^{\mathsf{f}}_k + s^{\mathsf{f}}_k f^{\mathsf{f,rq}}_k \right)$$

$$\text{s. t.} \quad s^{\mathsf{m}}_k \left(\frac{b^{\mathsf{out,f}}_k}{d_k} + \frac{(c^{\mathsf{co,f}}_k + c^{\mathsf{de,u}}_k)}{f^{\mathsf{f}}_k} \right) \leq \nu_{k,0}, \tag{6.20a}$$

$$\sum_{k\in\mathcal{B}} \left(s^{\mathsf{m}}_k d_k + s^{\mathsf{c}}_k d^{\mathsf{rq}}_k \right) \leq D^{\mathsf{max}}, \tag{6.20b}$$

$$(6.12a), (6.12b), (6.13),$$

where $\nu_{k,0} = \min\{(\eta - \Xi_{k,1})/w^{\mathsf{T}}_k, T^{\mathsf{max}}_k - T_{k,1}\} + (c_{k,1} + c^{\mathsf{de}}_k)/f^{\mathsf{f,rq}}_k - T^{\mathsf{c}}$, and $\Xi_{k,1}$ and $T_{k,1}$ are the optimal values of Ξ_k and T_k in $(\mathcal{P}_3)_k$, respectively; (6.20a)is determined by the time delay constraint as $\check{T}_k \leq \min(T^{\mathsf{max}}_k, (\eta - w^{\mathsf{E}}_k \xi_k)/w^{\mathsf{T}}_k)$ which is equivalent to constraints (6.16a) and (6.14). This constraint captures the fact that an application should be offloaded to the cloud server if the resulting WEDC is smaller than that achieved when the application is executed at the fog server and the delay constraint (6.14) is not violated. Because $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ is a difficult MINLP problem, we tackle it by reducing the set of variables based on the results in the following three propositions. In particular, Propositions 6.3–6.5 are introduced to respectively rewrite variables f^{f}_k, ω^{f}_k, and d_k, for all k as functions of the remaining variables. Subsequently, two algorithms are proposed to solve for the remaining variables, one based on a one-dimensional search of the Lagrangian multiplier, and the other one based on an iterative update of the Lagrangian multiplier.

Proposition 6.3. *For any value of d_k's satisfying (6.20b), the optimal solution of f^{f}_k in $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ can be determined as* $f^{\mathsf{f}\star}_k = s^{\mathsf{m}}_k \frac{(c^{\mathsf{co,f}}_k + c^{\mathsf{de,u}}_k)}{\nu_{k,0} - b^{\mathsf{out,f}}_k/d_k} = s^{\mathsf{m}}_k \mathcal{H}_0\Big(\omega^{\mathsf{f}}_k, d_k\Big)$, *where* $\mathcal{H}_0\Big(\omega^{\mathsf{f}}_k, d_k\Big) = \frac{\omega^{\mathsf{f}}_k d_k \left[\tilde{\gamma}^{\mathsf{co,f}}_{k,1} (\omega^{\mathsf{f}}_k)^{\gamma^{\mathsf{co,f}}_{k,2}} + \tilde{\gamma}^{\mathsf{co,f}}_{k,3}\right]}{\nu_{k,0}\omega^{\mathsf{f}}_k d_k - b^{\mathsf{in}}_k}$, $\tilde{\gamma}^{\mathsf{co,f}}_{k,1} = \gamma^{\mathsf{f}}_{k,0}\gamma^{\mathsf{co,f}}_{k,1}$, *and* $\tilde{\gamma}^{\mathsf{co,f}}_{k,3} = \gamma^{\mathsf{f}}_{k,0}\gamma^{\mathsf{co,f}}_{k,3} + c^{\mathsf{de,u}}_k$.

Proof. When $s^{\mathsf{m}}_k = 1$, the left-hand side of (6.20a) is inversely proportional to f^{f}_k; thus, f^{f}_k is minimized if users spend the maximum possible resources. □

[9]We note that by reducing the number of optimization variables in $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$, the complexity of the resulting algorithms for feasibility verification of $(\mathcal{P}^{\mathsf{ext}}_{\mathcal{B}})$ is lower than that of the PLA based algorithm.

Proposition 6.4. *When $s_k^{\mathsf{m}} = 1$ and $d_k \geq \bar{d}_{k,1}$, the optimal value of ω_k^{f}, denoted as $\omega_k^{\mathsf{f}\star}$, is given as follows:*

$$\omega_k^{\mathsf{f}\star} = \begin{cases} \omega_k^{\mathsf{max,f}}, \ \textit{if } \gamma_{k,2}^{\mathsf{co,f}} \leq 0 \cup \{\gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,1} < d_k \leq \bar{d}_{k,2}\}, \\ \text{inv}\Big(\mathcal{H}_1\Big(d_k\Big)\Big), \quad \textit{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, \bar{d}_{k,2} < d_k \leq \bar{d}_{k,3}, \\ \omega_k^{\mathsf{f,min}}, \quad \textit{if} \gamma_{k,2}^{\mathsf{co,f}} \geq 0, d_k > \bar{d}_{k,3}, \end{cases} \tag{6.21}$$

where $\bar{d}_{k,1} = b_k^{\mathsf{in}}/(\nu_{k,0}\omega_k^{\mathsf{f}})$, $\bar{d}_{k,2} = \mathcal{H}_1\Big(\omega_k^{\mathsf{max,f}}\Big)$, $\bar{d}_{k,3} = \mathcal{H}_1\Big(\omega_k^{\mathsf{f,min}}\Big)$, and $\text{inv}\Big(\mathcal{H}_1\Big(d_k\Big)\Big)$ is the value of ω_k^{f} for which $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)$ is equal to d_k, and $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big) \triangleq \frac{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} b_k^{\mathsf{in}} (\gamma_{k,2}^{\mathsf{co,f}}+1)\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}} b_k^{\mathsf{in}}}{\tilde{\gamma}_{k,1}^{\mathsf{co,f}} \nu_{k,0} \gamma_{k,2}^{\mathsf{co,f}} \Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}+1}}$.

Proof. The proof is given in Appendix 6.7.3. □

Based on the results in Propositions 6.3 and 6.4, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ is equivalent to the following problem:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}}) \quad \min_{\tilde{\Omega}_4} \quad \sum_{k\in\mathcal{B}} \Big[s_k^{\mathsf{m}} \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big) + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \Big]$$
$$\text{s. t. } (6.12a), (6.12b), (6.20b),$$

where $\tilde{\Omega}_4 = \cup_{k\in\mathcal{B}} \{s_k^{\mathsf{c}}, s_k^{\mathsf{f}}, s_k^{\mathsf{m}}, d_k\}$.

Proposition 6.5. *The optimal value of d_k for $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, denoted as $d_k^\star$, is given as follows:*

$$d_k^\star = \begin{cases} 0, \quad \textit{if} \quad s_k^{\mathsf{f}\star} = 1, \\ d_k^{\mathsf{rq}}, \quad \textit{if} \quad s_k^{\mathsf{c}\star} = 1, \\ \left\{ d_{k,\lambda} \middle| \left(\frac{\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)}{\partial d_k}\Big|_{d_k = d_{k,\lambda}} \right) + \lambda = 0 \right\}, \textit{otherwise}, \end{cases} \tag{6.22}$$

where λ is the Lagrange multiplier of constraint (6.20b).

Proof. The Lagrangian of problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ can be expressed as $\mathcal{L}(\tilde{\Omega}_4, \lambda) = \sum_{k\in\mathcal{B}} \Big[s_k^{\mathsf{m}} \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big) + s_k^{\mathsf{f}} f_k^{\mathsf{f,rq}} \Big] + \lambda \Big(\sum_{k\in\mathcal{B}} \Big[s_k^{\mathsf{m}} d_k + (1 - s_k^{\mathsf{f}} - s_k^{\mathsf{m}}) d_k^{\mathsf{rq}} \Big] - D^{\mathsf{max}} \Big)$. When $s_k^{\mathsf{m}\star} = 1$, the necessary conditions for the optimal solution $f_k^{\mathsf{f}\star}, d_k^\star$ can be obtained by setting the derivatives of $\mathcal{L}$ with respect to these

Algorithm 6.4. One-dimensional Search Based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **initialize**: Δ_λ, $\lambda = 0$, Assign $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible.
2: Define $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all k as in Step 2 and Step 3 of Algorithm 6.2.
3: **repeat**
4: Assign $\lambda = \lambda + \Delta_\lambda$. Compute $d_{k,\lambda}$ as in (6.22) and solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ to find $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$.
5: **if** $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \leq F^{\mathsf{f,max}}$ **then**
6: Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible; **break**
7: **end if**
8: **until** $\lambda = \lambda^{\mathsf{max}}$

variables equal to zero as follows:

$$\frac{\partial \mathcal{L}}{\partial d_k} = s_k^{\mathsf{m}} \left(\frac{\partial \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big)}{\partial d_k} + \lambda \right) = 0, \tag{6.23}$$

$$\lambda \left(\sum_{k \in \mathcal{B}} \Big[s_k^{\mathsf{m}} d_k + (1 - s_k^{\mathsf{f}} - s_k^{\mathsf{m}}) d_k^{\mathsf{rq}} \Big] - D^{\mathsf{max}} \right) = 0. \tag{6.24}$$

Based on (6.23), it can be verified that $d_k^\star$ can be expressed as in (6.22). □

Lemma 6.2. *The gradient $\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k$ is a monotonically increasing function of d_k.*

Proof. The proof is given in Appendix 6.7.4 . □

As can be verified, if $\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k \Big|_{d_k = \bar{d}_{k,1}} + \lambda > 0$, then $d_k^\star = d_{k,\lambda} = 0, s_k^{\mathsf{f}\star} = 1$ will be the optimal solution. When $s_k^{\mathsf{m}\star} = 1$, λ must be positive because $\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k$ is negative for all d_k. With the results in Lemma 6.2, we can conclude that for a given λ, there exists at most one value of d_k satisfying $\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k + \lambda = 0$. This means if the optimal λ is known, problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ can be solved effectively. Therefore, as described in the following, to solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$, we propose two algorithms: one is based on a one-dimensional search for λ, and the other one is based on iterative updating λ.

6.4.2.1 One-dimensional λ-search based two-stage algorithm (OSTS Alg.)

For a given λ, suppose that $d_{k,\lambda}$ satisfies $\partial \mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_k)/\partial d_k \Big|_{d_k = d_{k,\lambda}} + \lambda = 0$. By defining $f_{k,\lambda} = \mathcal{H}_0\Big(\omega_k^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k = d_{k,\lambda}}$, $\mu_{k,\lambda} = s_k^{\mathsf{m}}$, $\mu_{k,\lambda} = 1 - s_k^{\mathsf{c}}$, and $\mu_{k,\lambda} = s_k^{\mathsf{c}}(1 - x_k)$, we can find the optimal

solution of $\cup_{k\in\mathcal{B}}\{s_k^{\mathsf{c}}, x_k, d_k\}$ by solving the following problem:

$$\begin{aligned}(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda \quad & \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) = \min_{\cup_{k\in\mathcal{B}} s_{k,\lambda}} \sum_{k\in\mathcal{B}} \left[s_{k,\lambda}^{\mathsf{m}} f_{k,\lambda} + s_{k,\lambda}^{\mathsf{f}} f_k^{\mathsf{f,rq}} \right] \\ \text{s. t.} \quad & \sum_{k\in\mathcal{B}} s_{k,\lambda}^{\mathsf{m}} d_{k,\lambda} + (1 - s_{k,\lambda}^{\mathsf{f}} - s_{k,\lambda}^{\mathsf{m}}) d_k^{\mathsf{rq}} \le D^{\mathsf{max}}, \\ & s_{k,\lambda}^{\mathsf{m}}, s_{k,\lambda}^{\mathsf{f}} \in \{0,1\},\end{aligned}$$

where $s_{k,\lambda} = \{s_{k,\lambda}^{\mathsf{f}}, s_{k,\lambda}^{\mathsf{m}}\}$. The above transformed problem is an integer linear programming (ILP) problem, which can be solved effectively by CVX. Let $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$ be the optimum of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$, then we can find the optimum of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ as $\tilde{G}_{\mathcal{B},\eta}^{\mathsf{OSTS}\star} = \min_\lambda \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$. Moreover, it can be shown that when we increase λ, all $d_{k,\lambda}$ will decrease. Therefore, the maximum value of λ is λ^{max} satisfying $\mathcal{H}_0(\omega_k^{\mathsf{f}}, d_{k,\lambda^{\mathsf{max}}}) \ge f_k^{\mathsf{f,rq}}$, $\forall k \in \mathcal{B}$ and $\sum_{k\in\mathcal{B}} d_{k,\lambda^{\mathsf{max}}} \le D^{\mathsf{max}}$. Note that we can stop the search process when there exists a λ such that $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \le F^{\mathsf{f,max}}$. When the bisection search for η converges, we can find the optimum $\lambda^\star = \text{argmin}_\lambda\, \tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$, and the optimal variables $s_k^{\mathsf{m}\star} = s_{k,\lambda^\star}^{\mathsf{m}}$, $s_k^{\mathsf{f}\star} = s_{k,\lambda^\star}^{\mathsf{f}}$, $s_k^{\mathsf{c}\star} = 1 - s_k^{\mathsf{m}\star} - s_k^{\mathsf{f}\star}$, $f_k^{\mathsf{f}\star} = s_{k,\lambda^\star}^{\mathsf{m}} f_{k,\lambda^\star} + s_{k,\lambda^\star}^{\mathsf{f}} f_k^{\mathsf{f,rq}}$, and $d_k^\star = s_{k,\lambda^\star}^{\mathsf{m}} d_{k,\lambda^\star} + (1 - s_{k,\lambda^\star}^{\mathsf{f}} - s_{k,\lambda^\star}^{\mathsf{m}}) d_k^{\mathsf{rq}}, \forall k \in \mathcal{B}$. The OSTS algorithm for feasibility verification of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is summarized in Algorithm 6.4.

6.4.2.2 Iterative λ-update based two-stage algorithm (IUTS Alg.)

This method can solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ with very low complexity via Lagrangian dual updates. Specifically, the dual function of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ can be defined as $\mathcal{G}^{\mathsf{o}}(\lambda) = \min_{\tilde{\Omega}_4} \mathcal{L}(\tilde{\Omega}_4, \lambda)$, and the dual problem can be stated as

$$\max_\lambda \mathcal{G}^{\mathsf{o}}(\lambda) \quad \text{s. t.} \quad \lambda \ge 0. \tag{6.25}$$

Since the dual problem is always convex, $\mathcal{G}^{\mathsf{o}}(\lambda)$ can be maximized by using the standard sub-gradient method where the dual variable λ is iteratively updated as follows: $\lambda_n = \Bigg[\lambda_{n-1} + \delta_n \Big(\sum_{k\in\mathcal{B}} \Big(s_{k,\lambda_{n-1}}^{\mathsf{m}} d_{k,\lambda_{n-1}} + s_{k,\lambda_{n-1}}^{\mathsf{c}} d_k^{\mathsf{rq}} \Big) - D^{\mathsf{max}} \Big) \Bigg]^+$, where n denotes the iteration index, δ_n represents the step size, and $[a]^+$ is defined as $\max(0, a)$. The sub-gradient method is guaranteed to converge to the optimal value of λ for an initial primal point Ω_4 if the step size δ_n is chosen appropriately, e.g., $\delta_n \to 0$ when $n \to \infty$, which is met by setting $\delta_n = 1/\sqrt{n}$.

For a given λ_n, we can determine the primal variable $d_{k,\lambda_n} = \text{inv}(\mathcal{H}_2(\lambda_n))$. For given λ_n and d_{k,λ_n}, the primal problem becomes a linear program in $s_{k,\lambda_n}, \forall k \in \mathcal{B}$, which can be solved effectively by using standard linear optimization techniques. Moreover, the vertices in this problem are the points where the $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s are either 0 or 1. Thus, *solving the relaxed problem will also return binary values* 0 *or* 1. However, once the $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s take values of 0 or 1, the decision on the application execution location (fog or cloud) may be trapped at a local optimal solution such that the required fog computing resources cannot be updated to improve the solution. To overcome this critical issue, the gradient projection method can be adopted to slowly update variables $s^{\mathsf{m}}_{k,\lambda_n}$'s, $s^{\mathsf{f}}_{k,\lambda_n}$'s, and $s^{\mathsf{c}}_{k,\lambda_n}$'s as $\mathbf{s}_k^{(n+1)} = \mathbb{P}_{\Phi_k}\left(\mathbf{s}_k^{(n)} - \check{\delta}\nabla\mathbf{s}_k^{(n)}\right)$, where $\mathbf{s}_k^{(n)} = [s^{\mathsf{m}}_{k,\lambda_n}, s^{\mathsf{f}}_{k,\lambda_n}, s^{\mathsf{c}}_{k,\lambda_n}]$, $\check{\delta}$ is the step size, $\nabla\mathbf{s}_k^{(n)} = [\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_{k,\lambda_n}) + \lambda_n d_{k,\lambda_n}, \lambda_n f_k^{\mathsf{f,rq}}, \lambda_n d_k^{\mathsf{rq}}]$, and $\mathbb{P}_{\Phi_k}(.)$ is the projection onto the set $\Phi_k = \left\{\mathbf{s}_k | \mathbf{s}_k \geq 0,\ s^{\mathsf{f}}_{k,\lambda_n} + s^{\mathsf{c}}_{k,\lambda_n} + s^{\mathsf{m}}_{k,\lambda_n} \leq 1\right\}$. Finally, it can be verified that this iterative mechanism always converges [30].

6.4.3 Complexity Analysis

The overall complexity of the PLA algorithm for solving the extended problem $(\mathcal{P}_2^{\mathsf{ext}})$ is

$$\log_2\left(\frac{\eta^{\mathsf{max}} - \eta^{\mathsf{min}}}{\epsilon}\right)\left(K\mathcal{O}((\mathcal{P}_3)_k) + LK\mathcal{O}((\mathcal{P}_4)_k) + \mathcal{O}(\mathcal{P}^{\mathsf{PLA}}_{\mathsf{FV},\eta})\right), \text{i.e.,} |\mathcal{B}| \leq K.$$

Moreover, for given $\mathcal{B}$, $(\mathcal{P}^{\mathsf{PLA}}_{\mathsf{FV},\eta})$ is an NP-hard problem, solving it via an optimal exhaustive search entails a complexity of $\mathcal{O}(2^{(L+1)^{|\mathcal{B}|}})$, which is upper bounded by $\mathcal{O}(2^{(L+1)^K})$.

The proposed two-stage IUTS and OSTS algorithms for solving the extended problem have an overall complexity of $\log_2(\frac{\eta^{\mathsf{max}} - \eta^{\mathsf{min}}}{\epsilon})\left(K\mathcal{O}((\mathcal{P}_3)_k) + K\mathcal{O}((\mathcal{P}_4)_k) + \mathcal{O}(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})\right)$, i.e., $|\mathcal{B}| \leq K$. In Section 6.4.2.1, for given $\mathcal{B}$, problem $(\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda$ can be transformed to a standard knapsack problem as in [29], while the optimal d_k and ω_k can be computed directly for a given value of λ. Therefore, the complexity of Algorithm 6.4 to solve $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ by the OSTS method is $\mathcal{O}(\frac{\lambda^{\mathsf{max}}}{\Delta_\lambda}\nu_3|2\mathcal{B}|)$, where ν_3 is determined by the coefficients in $(\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda$ [29]. For the IUTS algorithm presented in Section 6.4.2.2, we can directly update $\lambda_n, d_{k,\lambda_n}, \mu_{k,i,\lambda_n}, \forall i,k,n$; which means that $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ has a complexity of $\mathcal{O}(N|\mathcal{B}|)$, where N is the number of iterations. We note that $\mathcal{O}((\mathcal{P}_3)_k)$ and $\mathcal{O}((\mathcal{P}_4)_k)$ are given in Section 6.3.6.

6.5 Numerical Results

6.5.1 Simulation Setup

We consider a hierarchical fog-cloud system consisting of K=10 users (except for Fig. 6.9) where the users are randomly and uniformly distributed in the cell coverage area with a radius of 800 m and the BS is located at the cell center. The simulation parameters provided in Table 6.1 are adopted, unless specified otherwise. Particularly, the path-loss is calculated as $\beta_k(\text{dB}) = 128.1 + 37.6\log_{10}(dist_k)$, where $dist_k$ is the geographical distance between user k and the BS (in km) [27]. We further set the beamforming gain as $M_0 = 5$, the maximum transmission bandwidth as $\rho_k^{\max} = 1$ MHz, and the noise power density as $\sigma_{\mathsf{bs}} = 1.381 \times 10^{-23} \times 290 \times 10^{0.9}$ W/Hz [31]. All users are assumed to have the same maximum clock speed of 2.4 GHz, a maximum transmit power of $p_k^{\max} = 0.22$ W, and the circuit power consumption per Hz is set to $p_{k,0} = 22$ nW/Hz. We assume that the number of transmission bits incurred to support computation offloading b_k^{in} is the same for all users.

Table 6.1 – Simulation Parameter Settings

Parameter	Setting
Path loss, β_k	$128.1 + 37.6\log_{10}(dist_k(\text{km}))$ [27]
Cell radius	800 meters [132]
Noise power density, σ_{bs}	3.18×10^{-20} W/Hz
Number of users K	10
Beamforming gain M_0	5
Max. transmission bandwidth $\rho_k^{\max}$	1 MHz
Max. delay time $T_k^{\max}$	1 second [40]
Max. clock speed $F_k^{\max}$	2.4 GHz [26]
Max. transmit power $p_k^{\max}$	0.22 W [120]
Circuit power consumption per Hz	$p_{k,0} = 22$ nW/Hz [133]
User computation demand	$c_k \in [1.8 - 2.4]$ Gcycles
Offloadable load	$c_{k,1} = 0.9c_k$ [134, 135]
Energy coefficient α_k	0.1×10^{-27} [35]
Time T^{c}	0.2 second
Raw data size b_k^{in}	4 Mbits
Max. fog computing resource $F^{\mathsf{f},\max}$	15 GHz [38]
Max. backhaul capacity $D^{\max}$	20 Mbps [41]
User compression ratio range: ω_k^{u}	$[2.3, 2.9]$
Coefficient κ	50
Fog compression ratio range: ω_k^{f}	$[3.4, 11.2]$

Moreover, the computation demands of the 10 users $\{c_1, c_2, ..., c_9, c_{10}\}$ are set randomly in the range $1.8 - 2.4$ Gcycles while the maximum delay time is to $T_k^{\mathsf{max}} = 1$ second, the non-offloadable load is $c_{k,0} = 0.1c_k$, and the offloadable load is $c_{k,1} = 0.9c_k$ for all users. We also set the energy coefficient as $\alpha_k = 0.1 \times 10^{-27}$ and the computing time at the cloud server as $T^{\mathsf{c}} = T_k^{\mathsf{max}}/5$. For the data compression algorithm, we set the parameters according to the top-left sub-figure in Fig. 6.2 as follows: $\gamma_{k,1}^{\mathsf{co}} = 0.03 \times 2.6^{32.28}$, $\gamma_{k,2}^{\mathsf{co}} = 32.28$, $\gamma_{k,3}^{\mathsf{co}} = 0.3$, $\gamma_{k,1}^{\mathsf{de}} = 0.115$, $\gamma_{k,2}^{\mathsf{de}} = -0.9179$, $\gamma_{k,3}^{\mathsf{de}} = 0.046, \forall k$, $\omega_k^{\mathsf{u,min}} = 2.3$, and $\omega_k^{\mathsf{u,max}} = 2.9$. The energy and delay weights are chosen so that $w_k^{\mathsf{E}} + w_k^{\mathsf{T}} = 1, \forall k$. Simulation results are obtained by averaging over 100 realizations of the random locations of the users. Finally, for all figures, we set the raw data size as $b_k^{\mathsf{in}} = 4$ Mbits (except for Figs. 6.5, 6.7 and 6.9), $w_k^{\mathsf{E}} = 2w_k^{\mathsf{T}}, \forall k$ (except for Fig. 6.8), the maximum fog computing resource as $F^{\mathsf{f,max}} = 15$ GHz, the maximum backhaul capacity as $D^{\mathsf{max}} = 20$ Mbps (except for Figs. 6.7 and 6.8), and $\kappa = 50$ (except for Figs. 6.5 and 6.6), where κ captures the relationship between $\gamma_{k,0}^{\mathsf{u}}$ in (6.1) and the raw data size as $\gamma_{k,0}^{\mathsf{u}} = \kappa b_k^{\mathsf{in}}$ [32].

In practice, a fog server can support more powerful data compression algorithms compared to the users. This implies that the compression ratio for the fog server is much larger than that for the users. Therefore, when the fog server decompresses and re-compresses data, we set the parameters according to the top-middle sub-figure in Fig. 6.2 as follows: $\gamma_{k,1}^{\mathsf{co,f}} = 0.076, \gamma_{k,2}^{\mathsf{co,f}} = 0.7116, \gamma_{k,3}^{\mathsf{co,f}} = 0.5794$, $\omega_k^{\mathsf{f,min}} = 3.4$, and $\omega_k^{\mathsf{f,max}} = 11.2$. The step size is set as $\check{\delta} = 0.1$. For the proposed algorithms presented in Section 6.3 and Section 6.4, numerical results are shown in Figs. 6.5–6.9 and Figs. 6.10–6.12, respectively.

6.5.2 Results for Data Compression at only Mobile Users

In Fig. 6.5, we show the significant benefits of data compression for computation offloading where the min-max WEDC (called WEDC for brevity) vs. b_k^{in} is plotted for six different schemes: the 'Local-execution' scheme in which all users' applications are executed locally; the 'Alg. in [15] (w/o Comp)' scheme in which the benchmark algorithm in [15] is applied with $\omega_k^{\mathsf{u}} = 1, \forall k$, and no data compression [10]; the 'JCORA Alg. w/o Comp' in which the proposed JCORA algorithm is applied with $\omega_k^{\mathsf{u}} = 1, \forall k$, and no data compression (the other variables are optimized as in the JCORA

[10] As discussed in Section I, this paper provides the first study of joint data compression and computation offloading in hierarchical fog-cloud systems. Therefore, the recent work [15] on computation offloading in hierarchical fog-cloud systems, which does not exploit data compression, is selected as benchmark for performance comparison.

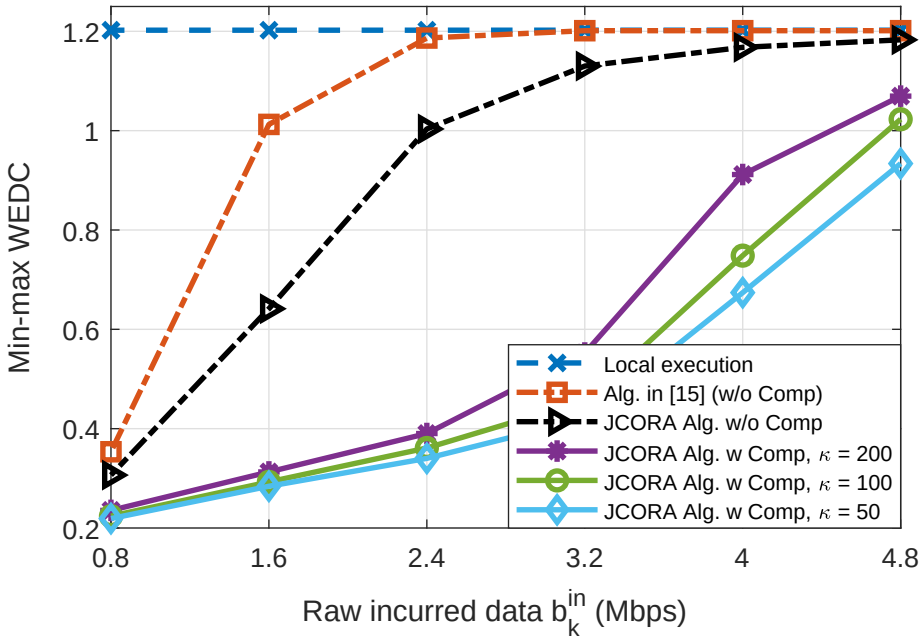

Figure 6.5 – Min-max WEDC vs. b_k^{in}.

algorithm); and three other instances of the proposed JCORA algorithm with data compression and three different values of $\kappa = 50, 100, 200$ ($\kappa = \gamma_{k,0}^{\text{u}}/b_k^{\text{in}}$). To guarantee a fair comparison between the 'Alg. in [15] (w/o Comp)' scheme and our proposed schemes, we also apply MIMO and optimize the offloading decision and the allocation of the fog computing resources, transmit power, bandwidth, and local CPU clock speed for the 'Alg. in [15] (w/o Comp)' scheme. In addition, for the remaining variable d_k, we allocate the backhaul capacity equally to the users that offload their tasks to the cloud server.

As can be observed from Fig. 6.5, computation offloading can greatly improve the WEDC when there are sufficient radio and computing resources to support the offloading (e.g., the incurred amount of data is not too large). Specifically, computation offloading even without data compression can result in a significant reduction of the WEDC compared to local execution, especially when the incurred amount of data b_k^{in} is small such that the constrained radio resources do not limit performance. Furthermore, even without exploiting data compression, our proposed algorithm (JCORA Alg. w/o Comp) results in a much better performance than the algorithm proposed in [15]. This is because our proposed design jointly optimizes the offloading decisions and the computing and radio resource allocation, while in [15], the offloading decisions are found nearly independent of the computing and radio resource allocation. In particular, the semidefinite relaxation technique employed in [15] may not always guarantee the rank-1 condition for the optimized matrix. Joint

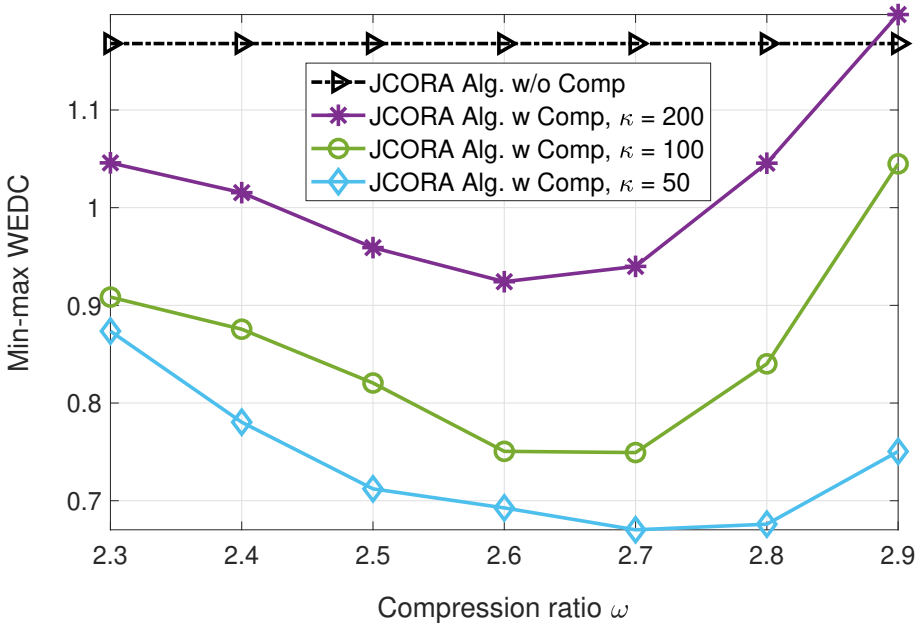

Figure 6.6 – Min-max WEDC vs. compression ratio.

optimization of data compression, computation offloading, and resource allocation can lead to a significant further reduction of the WEDC for a larger range of b_k^{in} (e.g., when $b_k^{\mathsf{in}} = 2.4$ Mbps, the min-max WEDC is reduced by up to 65%). However, the energy and time consumed for (de)compression also affect the achievable min-max WEDC, and their impact tends to become stronger for larger $\gamma_{k,0}^{\mathsf{u}}$ and when the available radio resource is more limited.

In Fig. 6.6, we investigate the impact of the compression ratio on the min-max WEDC for the JCOCA scheme with and without data compression for different values of $\omega_k^{\mathsf{u}} = \omega, \forall k$ (i.e., the compression ratio ω_k^{u} is fixed while the remaining variables are optimized as in the JCOCA scheme). As can be seen, there is an optimal ω that achieves the minimum WEDC. Moreover, the optimal value of ω tends to decrease for increasing computational load because the optimal compression ratio has to efficiently balance the demand on the radio and computing resources. In fact, for the right choice of ω, the *"JCORA Alg. w Comp"* scheme greatly outperforms the *"JCORA Alg. w/o Comp"* scheme. Moreover, this figure shows that for the optimal ω, 29% reduction in the min-max WEDC can be achieved compared to the worst choice of ω.

Fig. 6.7 shows the computational loads processed locally as well as in the fog and cloud servers when $b_k^{\mathsf{in}} = 4.8$ Mbits for four different scenarios: 1) $F^{\mathsf{f,max}} = 15$ GHz, $D^{\mathsf{max}} = 20$ Mbps; 2) $F^{\mathsf{f,max}} = 20$ GHz, $D^{\mathsf{max}} = 20$ Mbps; 3) $F^{\mathsf{f,max}} = 15$ GHz, $D^{\mathsf{max}} = 30$ Mbps; and 4) $F^{\mathsf{f,max}} = 20$ GHz, $D^{\mathsf{max}} = 30$ Mbps. The results shown in Fig. 6.7 suggest that more of the users' computational

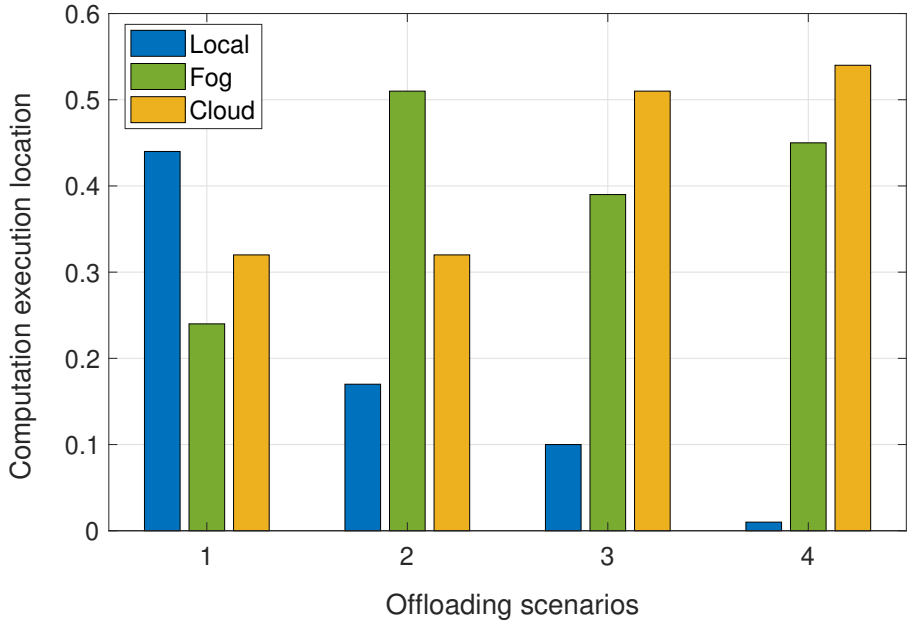

Figure 6.7 – User, fog, and cloud computational load processing.

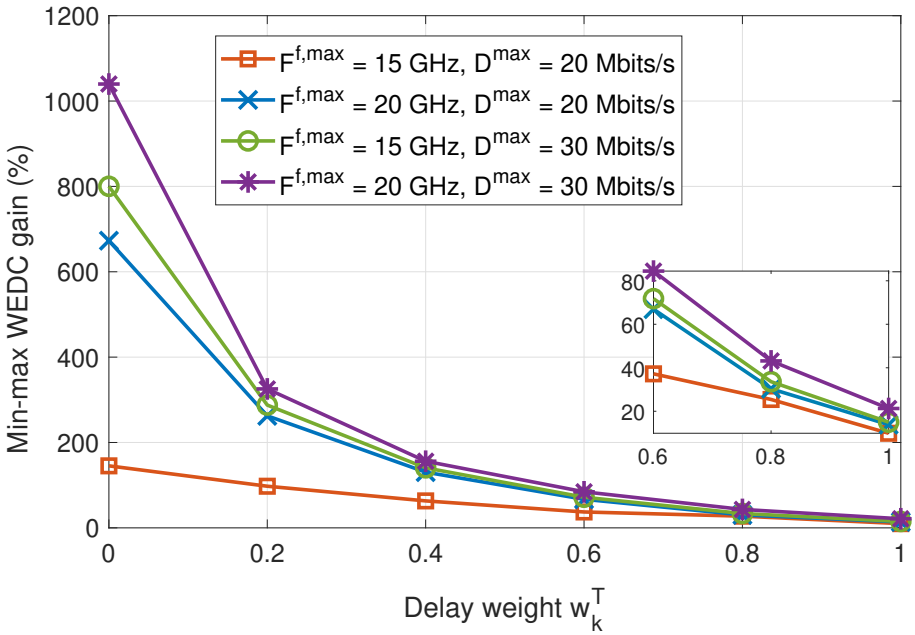

Figure 6.8 – Min-max WEDC gain vs. delay weight.

load should be offloaded and executed at the fog and cloud servers if sufficient resources to support the offloading process are available. Particularly, nearly all users offload their computation tasks in Scenario 4, while in Scenario 1, about half of the users offload their computation demand.

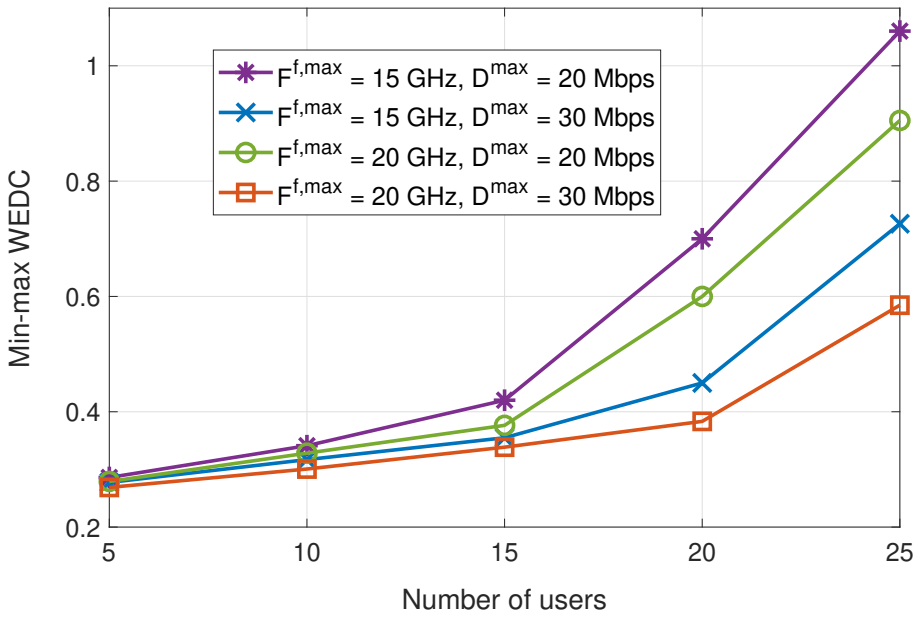

Figure 6.9 – Min-max WEDC vs. number of users.

In Fig. 6.8, we show the min-max WEDC gain due to data compression as a function of the delay weight w_k^T. The min-max WEDC gain is computed as $\frac{\eta^{\mathsf{NoComp}\star}-\eta^{\mathsf{Comp}\star}}{\eta^{\mathsf{Comp}\star}} \times 100$ (%) where $\eta^{\mathsf{Comp}\star}$ and $\eta^{\mathsf{NoComp}\star}$ denote the optimal min-max WEDCs with and without data compression under the JCORA framework. When energy saving is the only concern for the mobile devices ($w_k^{\mathsf{T}} = 0, w_k^{\mathsf{E}} = 1$), this figure confirms that JCORA with data compression can save more than 170% of energy compared with JCORA without data compression even for the scenario with $F^{\mathsf{f,max}} = 15$ GHz and $D^{\mathsf{max}} = 20$ Mbps. The min-max WEDC gain decreases when we focus more on latency (i.e., for higher delay weight w_k^T). Moreover, for $w_k^{\mathsf{T}} = 1$, data compression results in a 15% reduction of the execution delay for $F^{\mathsf{f,max}} = 15$ GHz, $D^{\mathsf{max}} = 20$ Mbps, and about 25% delay reduction for $F^{\mathsf{f,max}} = 20$ GHz, $D^{\mathsf{max}} = 30$ Mbps.

In Fig. 6.9, we show the min-max WEDC vs. the number of users in the system for $b_k^{\mathsf{in}} = 2.4$ Mbps, $\forall k$. When there are more users that may offload their computational loads to the fog and cloud servers, the available resources that can be allocated to each user become smaller; therefore, the min-max WEDC increases. However, the proposed JCORA scheme still achieves the optimal performance in the multi-user hierarchical fog-cloud system.

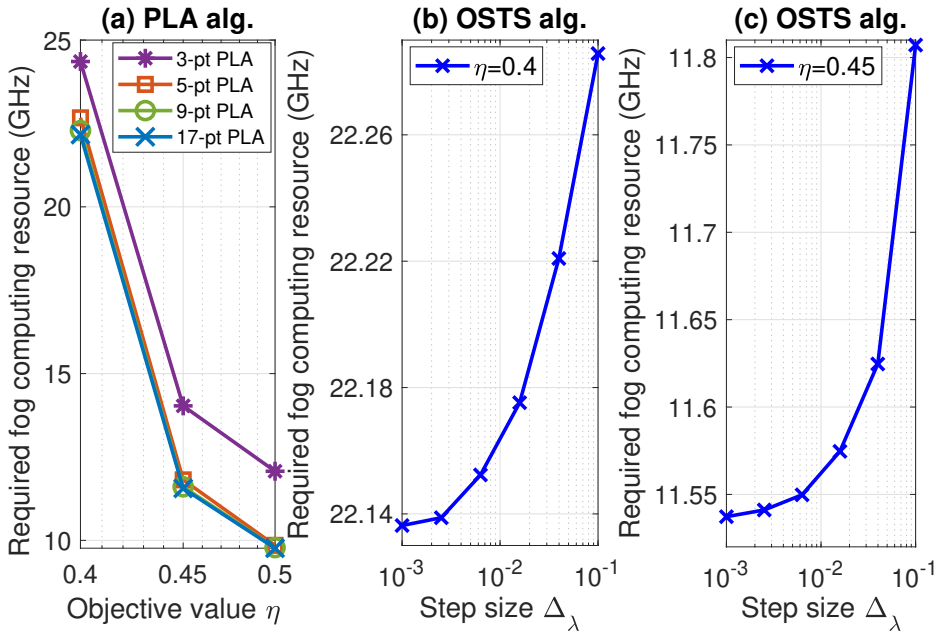

Figure 6.10 – Accuracy of proposed PLA and OSTS algs.

6.5.3 Results for Data Compression at both Mobile Users and Fog Server

To evaluate the system performance when data compression is performed at both the mobile users and the fog server, we consider the following parameter setting: $\gamma^{\mathsf{f}}_{k,0} = \gamma^{\mathsf{u}}_{k,0}$ (except for Fig. 6.12), $F^{\mathsf{f,max}} = 15$ GHz, and $D^{\mathsf{max}} = 20$ Mbps. In Fig. 6.10, we show the required computing resources for the proposed PLA and OSTS algorithms when solving the extended problem. In Fig. 6.10-(a), 'n-pt PLA' corresponds to the n-point PLA method. In the PLA method, when the number of points used to approximate the actual function is sufficiently large, the difference between the actual and approximated functions becomes negligible. As shown in Fig 6.10-(a), there is only a small difference in the required fog computing resources when the number of points increases from 5 to 9. In addition, these required resources are nearly identical for both the 9-point and 17-point curves. Therefore, we use '9-pt PLA' as a benchmark method to evaluate the performance of the OSTS and IUTS algorithms. The middle and right sub-figures illustrate the accuracy of the OSTS algorithm in solving problem ($\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta}$) vs. the step size Δ_λ. Specifically, these figures show that the value of $G^{\mathsf{OSTS}\star}_{\mathcal{B},\eta}$ becomes stable when Δ_λ is about 5×10^{-3}. Moreover, the value of $G^{\mathsf{OSTS}\star}_{\mathcal{B},\eta}$ achieved with the OSTS algorithm at $\Delta_\lambda = 5\times10^{-3}$ is almost the same as the value of $\hat{G}^{\mathsf{PLA}\star}_{\mathcal{B},\eta}$ achieved with '17-pt PLA', which means that the approximated problem ($\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta}$) can be used to find a close-to-optimal solution of the extended problem. Besides, the difference in $G^{\mathsf{OSTS}\star}_{\mathcal{B},\eta}$ for $\Delta_\lambda = 0.1$ and

$\Delta_\lambda = 0.001$ is less than 2%, which means that a large step size ($\Delta_\lambda = 0.1$) can be used to make the OSTS algorithm converge quickly while still guaranteeing good system performance.

The benefits of data re-compression at the fog are shown in Fig. 6.11 where we plot the min-max WEDC vs. b_k^{in} for four different schemes: the 'JCORA Alg. w Comp' scheme in which data are compressed only at the users while the three remaining schemes correspond to the proposed algorithms for the extended case. In particular, '9-pt PLA Alg. w Fog Comp', 'OSTS Alg. w Fog Comp', and 'IUTS Alg. w Fog Comp' correspond to the 9-point PLA, OSTS, and IUTS algorithms, respectively, which perform compression at both the users and the fog server. For $b_k^{\text{in}} = 4$ Mbits, an additional min-max WEDC reduction of 35% can be achieved by performing data compression at both the users and the fog server. Moreover, the required radio resources decrease with decreasing b_k^{in}; therefore, the gain is reduced due to the decreasing demand for data transmission. When b_k^{in} increases, the main bottleneck for computation offloading are the limited radio resources available to support data transmissions between the users and the fog server; therefore, the gain due to data re-compression at the fog server becomes less significant. This figure also confirms that the '9-pt PLA', 'OSTS', and 'IUTS' schemes achieve almost the same min-max WEDC.

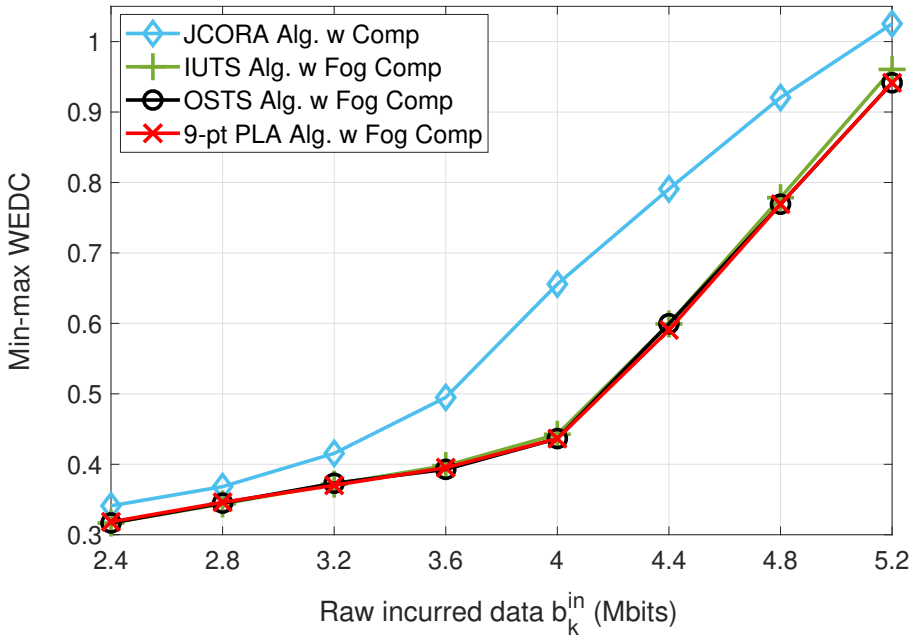

Figure 6.11 – Min-max WEDC in general design scenario.

In Fig. 6.12, we plot the min-max WEDC vs. the ratio between the maximum computational loads (in CPU cycles) required to compress data at the fog server ($\gamma_{k,0}^{\text{f}}$) and the user ($\gamma_{k,0}^{\text{u}}$) for

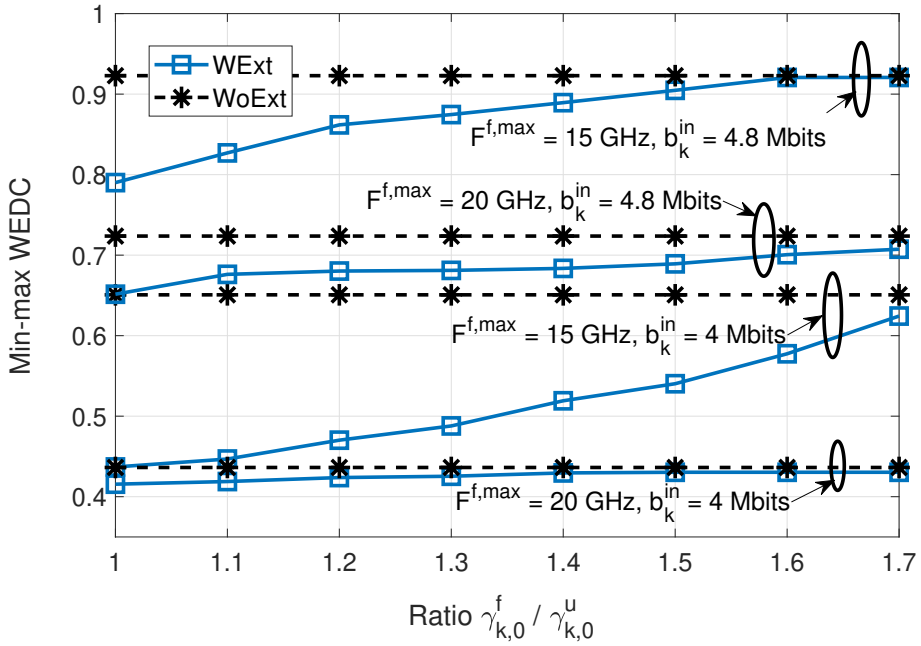

Figure 6.12 – Min-max WEDC vs. $\gamma^{\mathsf{f}}_{k,0}/\gamma^{\mathsf{u}}_{k,0}$.

different values of $F^{\mathsf{f,max}}$ and b^{in}_k. The 'WoExt' and 'WExt' correspond to the JCORA and OSTS algorithms presented in Sections 6.3 and 6.4, respectively. This figure shows that data compression at the fog server can bring additional performance benefits, especially in scenarios with limited fog computing resources (i.e., $F^{\mathsf{f,max}} = 15$ GHz). As the compression ratio adopted at the fog server could be much larger than that at the users, a better performance can be obtained by applying data compression at both the users and the fog server when $\gamma^{\mathsf{f}}_{k,0}$ is not much larger than $\gamma^{\mathsf{u}}_{k,0}$. Otherwise, if the cost due to data re-compression becomes larger, the benefits of adopting *Mode 3* are less significant (i.e., for $\gamma^{\mathsf{f}}_{k,0} = 1.7\gamma^{\mathsf{u}}_{k,0}$).

6.6 Conclusion

In this paper, we have proposed novel and efficient algorithms for joint data compression and computation offloading in hierarchical fog-cloud systems which minimize the weighted energy and delay cost while maintaining user fairness. Specifically, we have considered the cases where data compression is leveraged at only the mobile users and at both the mobile users and the fog server, respectively. Numerical results have confirmed the significant performance gains of the proposed algorithms compared to conventional schemes not using data compression. Particularly, the following key observations can be drawn from our numerical studies: 1) Joint data compression and

computation offloading can result in min-max WEDC reductions of up to 65% compared to optimal computation offloading without data compression; 2) the proposed JCORA scheme can efficiently distribute the computational load among the mobile users, the fog server, and the cloud server and exploits the available system resources in an optimal manner; 3) when energy saving is the only concern for the mobile users, the JCORA scheme can achieve an energy saving gain of up to a few hundred percent compared to optimal computation offloading without data compression; and 4) an additional min-max WEDC reduction of up to 35% can be achieved by further employing data compression at the fog server. In future work, we plan to extend our designs to multi-task offloading and systems with multiple fog servers.

6.7 Appendices

6.7.1 Proof of Theorem 6.1

Assume that $(\mathcal{A}', \mathcal{B}')$ is an optimal classification corresponding to the optimum value $\eta^\star$. Due to Statement 2 in Lemma 6.1 and Proposition 6.1, we have the following results:

$$\max(\eta_{\mathcal{A}'}, \eta_{\mathcal{B}'}) = \eta^\star, \quad \eta_{\mathcal{A}'} = \max_{k \in \mathcal{A}'} \eta_k^{\mathsf{lo}}. \tag{6.26}$$

If there is no user k in $\mathcal{B}$ whose η_k^{lo} is less than or equal to $\eta^\star$, we can conclude that $(\mathcal{A}', \mathcal{B}') \equiv (\mathcal{A}^\star, \mathcal{B}^\star)$. Then, $(\mathcal{A}^\star, \mathcal{B}^\star)$ must be an optimal classification.

Conversely, if there exists a user k in $\mathcal{B}$ such that $\eta_k^{\mathsf{lo}} \leq \eta^\star$, we will prove that the user classification determined in Theorem 6.1 is also an optimal classification. Let $\mathcal{C} = \{k \in \mathcal{B}' | \eta_k^{\mathsf{lo}} \leq \eta^\star\}$. Then, it is easy to see that $\mathcal{A}^\star = \mathcal{A}' \cup \mathcal{C}$ and $\mathcal{B}^\star = \mathcal{B}'/\mathcal{C}$. According to the definition of $\mathcal{C}$, (6.26), and the result in Proposition 6.1, we have $\eta_{\mathcal{A}^\star} \leq \eta^\star$. In addition, since $\mathcal{B}^\star \subset \mathcal{B}'$, because of Statement 3 in Lemma 6.1, we can conclude that $\eta_{\mathcal{B}^\star} \leq \eta_{\mathcal{B}'} \leq \eta^\star$. Using these results, we can conclude that $(\mathcal{A}^\star, \mathcal{B}^\star)$ is an optimal classification.

6.7.2 Proof of Proposition 6.2

Functions $\mathcal{Q}_{k,1}$ and $\mathcal{Q}_{k,2}$ are sums of exponential terms with positive coefficients; therefore, they are convex with respect to the variables in set $\tilde{\Omega}_{2,k}$ as proven in [1]. On the other hand, the first term of the WEDC and the total delay can be represented via function $\mathcal{H}(\tilde{p}_k, y_k) = \frac{a_{k,0}\mathrm{e}^{a_{k,1}\tilde{p}_k + a_{k,2}y_k}}{\log\left(1+\beta_{k,0}\mathrm{e}^{\tilde{p}_k}\right)}$, where $y_k \in \{\tilde{\omega}_k^{\mathsf{u}}, \tilde{\rho}_k, \tilde{l}_k\}$, $a_{k,0} > 0$, $a_{k,1} = \{0, 1\}$, and $\beta_{k,0}\mathrm{e}^{\tilde{p}_k} > 0$ due to the required positive data rate when users decide to offload their computational load.

Now, we will show that $\mathcal{H}(\tilde{p}_k, y_k)$ is a convex function of $\tilde{p}_k$ and y_k. Firstly, $\mathcal{H}(\tilde{p}_k, y_k)$ is convex with respect to y_k. Now, we need to prove that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 \geq 0$ and the determinant $|H(\tilde{p}_k, y_k)| > 0$, where $H(\tilde{p}_k, y)$ is the Hessian matrix of $\mathcal{H}(\tilde{p}_k, y_k)$.

Because we have $u_k = \beta_{k,0}\mathrm{e}^{\tilde{p}_k} > 0$ and the fact that $\log(1+u_k) < u_k, \forall u_k > 0$, it can be verified that $|H(\tilde{p}_k, y)| = \frac{a_{k,0}a_{k,2}^2\beta_{k,0}[u_k - \log(1+u_k)]\mathrm{e}^{(2a_{k,1}+1)\tilde{p}_k + 2a_{k,2}y}}{(1+u_k)^2\log^4(1+u_k)} > 0$. In addition, we have

$$\frac{\partial^2\mathcal{H}(\tilde{p}_k, y_k)}{\partial\tilde{p}_k^2} = \begin{cases} \frac{u_k[2u_k - \log(1+u_k)]}{(1+u_k)^2\log^3(1+u_k)}, & \text{if} \quad a_{k,1} = 0, \\ \frac{a_{k,0}\mathrm{e}^{a_{k,2}y}\mathrm{e}^{\tilde{p}_k}\mathcal{H}_{\mathrm{a}}(u_k)}{(1+u_k)^2\log^3(1+u_k)}, & \text{if} \quad a_{k,1} = 1, \end{cases} \tag{6.27}$$

where $\mathcal{H}_{\mathrm{a}}(u_k) = (1+u_k)^2\log^2(1+u_k) + 2u_k^2 - (3u_k + 2u_k^2)\log(1+u_k)$. From (6.27), it can be verified that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 > 0, \forall u_k > 0$ when $a_{k,1} = 0$. For the case with $a_{k,1} = 1$, since $\mathcal{H}_{\mathrm{a}}(u_k)$ is a quadratic function of $\log(1+u_k)$, the discriminant of $\mathcal{H}_{\mathrm{a}}(u_k)$ is $u_k^2\left[2 - \left(1+2u_k\right)^2\right]$, which leads to $\mathcal{H}_{\mathrm{a}}(u_k) = (1+u_k)^2\prod_{j=\{-1,1\}}\left(\log(1+u_k) - u_{k,j}\right)$ if $u_k \leq \frac{\sqrt{2}-1}{2}$, where $u_{k,j} = \frac{u_k(3+2u_k) + ju_k\sqrt{2-(1+2u_k)^2}}{2(1+u_k)^2}$, $j = \{-1, 1\}$. Otherwise, $\mathcal{H}_{\mathrm{a}}(u_k)$ will be positive. Using again $\log(1+u_k) < u_k, \forall u_k > 0$, we have $u_{k,\{1\}} - \log(1+u_k) \geq u_{k,\{-1\}} - \log(1+u_k) \geq u_{k,\{-1\}} - u_k > 0, \forall u_k > 0$. This implies that $\mathcal{H}_{\mathrm{a}}(u_k) > 0, \forall u_k > 0$, and we can conclude that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 > 0$ as shown in (6.27). As $\mathcal{H}(\tilde{p}_k, y_k)$ is a convex function, Ξ_k and T_k are also convex. Furthermore, $(6.5g)_k$ can be easily transformed to a linear constraint as $\tilde{\rho}_k + \tilde{p}_k \leq \log(P_k^{\mathsf{max}})$, while $(6.5b)_k$, $(6.5f)_k$, and $(6.5h)_k$ can be converted to box constraints for $\tilde{f}_k^{\mathsf{u}}$, $\tilde{\omega}_k^{\mathsf{u}}$, and $\tilde{\rho}_k$, respectively. Therefore, $(\mathcal{P}_3)_k$ is a convex optimization problem with respect to $\tilde{\Omega}_{2,k} \cup \tilde{l}_k$.

6.7.3 Proof of Proposition 6.4

We have the derivative $\partial\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)/\partial\omega_k^{\mathsf{f}}=\mathcal{H}_3(\omega_k^{\mathsf{f}},d_k)/(\nu_{k,0}\omega_k^{\mathsf{f}}d_k-b_k^{\mathsf{in}})^2$, where $\mathcal{H}_3(\omega_k^{\mathsf{f}},d_k)=d_k\Big[-\tilde{\gamma}_{k,1}^{\mathsf{co,f}}b_k^{\mathsf{in}}(\gamma_{k,2}^{\mathsf{co,f}}+1)\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}-\tilde{\gamma}_{k,3}^{\mathsf{co,f}}b_k^{\mathsf{in}}+\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\nu_{k,0}\gamma_{k,2}^{\mathsf{co,f}}d_k\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}+1}\Big]$. As $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)$ is positive when $s_k^{\mathsf{m}}=1$, it implies that $\nu_{k,0}\omega_k^{\mathsf{f}}d_k>b_k^{\mathsf{in}}$. Therefore, we can infer that $\mathcal{H}_3(\omega^{\mathsf{f}},d_k)\leq-\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\Big(\omega^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}b_k^{\mathsf{in}}d_k-\tilde{\gamma}_{k,3}^{\mathsf{co,f}}b_k^{\mathsf{in}}d_k<0,\forall\omega^{\mathsf{f}},d_k$ if $\gamma_{k,2}^{\mathsf{co,f}}\leq 0$. Hence, $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)$ achieves its minimal value at $\omega_k^{\mathsf{f}\star}=\omega_k^{\mathsf{max,f}}$ when $\gamma_{k,2}^{\mathsf{co,f}}\leq 0$. When $\gamma_{k,2}^{\mathsf{co,f}}>0$, it can be verified that $\mathcal{H}_3\Big(\omega_k^{\mathsf{f}\star},d_k\Big)=0$ if and only if $d_k=\mathcal{H}_1\Big(\omega_k^{\mathsf{f}\star}\Big)$. On the other hand, the derivative of $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)$ is $\frac{\partial\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)}{\partial\omega_k^{\mathsf{f}}}=-\frac{(\gamma_{k,2}^{\mathsf{co,f}}+1)}{\gamma_{k,2}^{\mathsf{co,f}}}\frac{b_k^{\mathsf{in}}(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}+\tilde{\gamma}_{k,3}^{\mathsf{co,f}}))}{\nu_{k,0}\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}+2}}<0$. So, $\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)$ is a monotonically decreasing function with respect to ω_k^{f}. Therefore, $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)$ is minimized if $\omega_k^{\mathsf{f}}=\omega_k^{\mathsf{f}\star}$ satisfies (6.21).

6.7.4 Proof of Lemma 6.2

First, it can be verified that $\frac{\partial\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)}{\partial d_k}=-\frac{b_k^{\mathsf{in}}\omega_k^{\mathsf{f}}\Big[\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}+\tilde{\gamma}_{k,3}^{\mathsf{co,f}}\Big]}{\Big(\nu_{k,0}\Big(\omega_k^{\mathsf{f}}\Big)d_k-b_k^{\mathsf{in}}\Big)^2}=\mathcal{H}_2\Big(\omega_k^{\mathsf{f}},d_k\Big)$. As $\partial\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)/\partial\omega_k^{\mathsf{f}}<0$ for all ω_k^{f}, $\omega_k^{\mathsf{f}\star}$ will not increase when $d_k>\bar{d}_{k,1}$ increases. When $\gamma_{k,2}^{\mathsf{co,f}}\leq 0$, $\omega_k^{\mathsf{f}\star}=\omega_k^{\mathsf{max,f}}$ as proved in Proposition 6.4. Therefore, $\mathcal{H}_2\Big(\omega_k^{\mathsf{max,f}},d_k\Big)$ increases with respect to d_k. When $\gamma_{k,2}^{\mathsf{co,f}}>0$, we will show that $\mathcal{H}_2\Big(\omega_{k,1}^{\mathsf{f}\star},d_k\Big)\Big|_{d_k=d_{k,1}}<\mathcal{H}_2\Big(\omega_{k,2}^{\mathsf{f}\star},d_k\Big)\Big|_{d_k=d_{k,2}}$, where $\bar{d}_{k,1}<d_{k,1}<d_{k,2}$ and $\omega_{k,i}^{\mathsf{f}\star}$ denotes the optimal value of ω_k^{f} when d_k is equal to $d_{k,i}$, for $i=1,2$.

Indeed, when ω_k^{f} is fixed, $\mathcal{H}_2\Big(\omega_k^{\mathsf{f}},d_k\Big)$ is an increasing function of d_k. The second derivative of $\mathcal{H}_0\Big(\omega_k^{\mathsf{f}},d_k\Big)$ when substituting $d_k=\mathcal{H}_1\Big(\omega_k^{\mathsf{f}}\Big)$ is given as $\frac{\partial\mathcal{H}_2\Big(\omega_k^{\mathsf{f}},d_k\Big)}{\partial\omega_k^{\mathsf{f}}}=-\frac{\mathcal{H}_4\Big(\omega_k^{\mathsf{f}}\Big)}{\Big(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}+\tilde{\gamma}_{k,3}^{\mathsf{co,f}}\Big)^2}$, where $\mathcal{H}_4\Big(\omega_k^{\mathsf{f}}\Big)=(\tilde{\gamma}_{k,1}^{\mathsf{co,f}})^2\Big(\gamma_{k,2}^{\mathsf{co,f}}\Big)^2\Big(\omega_k^{\mathsf{f}}\Big)^{2\gamma_{k,2}^{\mathsf{co,f}}}\Big(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\Big(\gamma_{k,2}^{\mathsf{co,f}}+1\Big)\Big(\omega_k^{\mathsf{f}}\Big)^{\gamma_{k,2}^{\mathsf{co,f}}}+\tilde{\gamma}_{k,3}^{\mathsf{co,f}}\Big(2\gamma_{k,2}^{\mathsf{co,f}}+1\Big)\Big)>0$, for all ω_k^{f} when $\gamma_{k,2}^{\mathsf{co,f}}>0$. Thus, it can be concluded that $\mathcal{H}_2\Big(\omega_k^{\mathsf{f}},d_k\Big)$ is a decreasing function of ω_k^{f}. Furthermore, the optimal solution $\omega_k^{\mathsf{f}\star}$ monotonically decreases as d_k increases as shown

in (6.21); hence, $\omega_{k,1}^{\mathsf{f}\star} \geq \omega_{k,2}^{\mathsf{f}\star}$. Therefore, we have $\mathcal{H}_2\Big(\omega_{k,1}^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k=d_{k,1}} \leq \mathcal{H}_2\Big(\omega_{k,2}^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k=d_{k,1}} < \mathcal{H}_2\Big(\omega_{k,2}^{\mathsf{f}\star}, d_k\Big)\Big|_{d_k=d_{k,2}}$.

Chapter 7

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

The content of this chapter was published in IEEE Communications Letter in the following paper:

Ti Ti Nguyen, Vu Nguyen Ha, and Long Bao Le, "Wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks," *IEEE Commun. Letters,* vol. 24, no. 2, pp. 410-413, Feb. 2020.

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.

7.1 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) [3]. 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 [24, 136, 137]. 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 or 2, 4, 8, 16 times of 180 kHz and the time duration equal or 2, 4, 8, 16 times smaller than 0.5 ms considering 7 OFDM symbols per mini-slot. The scheduling problem for resource allocation (RA) of two-dimensional (2D) time-frequency resources has been studied in several recent works [88, 89]. However, the 5G-NR mixed numerology has not been fully studied in [88] while [89] assumes that only two different numerologies exist in the system. Finally, the authors of [4] 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 Iterative Greedy Algorithm (IGA), are proposed to tackle the underlying problem. Because the complexity of solving an integer linear programming (ILP) problem can be decreased exponentially if we decrease the number of optimization variables, the RPA algorithm is developed by partitioning the resources and users into smaller groups, optimizing the RA for each group, then performing resource defragmentation and additional resource assignment for unallocated resources to obtain a final solution. In the RPA algorithm, the RA subproblem for each group can be solved efficiently by using the Gurobi solver. In the IGA algorithm, we iteratively allocate PRBs to users based on an appropriate resource assignment weight to obtain an efficient scheduling solution with low computation complexity. Numerical studies are performed to demonstrate the efficacy of the proposed algorithms.

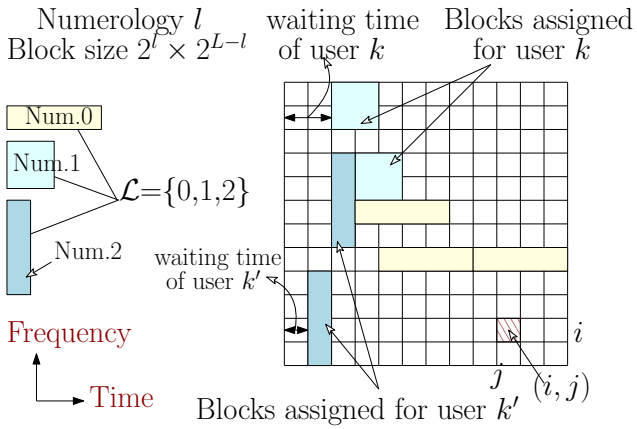

Figure 7.1 – PRB allocation in 5G wireless networks with mixed numerology

7.2 System Model

We consider the 5G system where the available time-frequency resource is divided into resource elements (RE). Each RE occupies the bandwidth of $\Delta^{\mathsf{f}}_{\mathsf{min}}$ (Hz) and the slot duration of $\Delta^{\mathsf{t}}_{\mathsf{min}}$ (seconds). The link/channel conditions for each subcarrier are assumed unchanged during the scheduling time. Moreover, we assume that the 2D RA is performed over each scheduling interval of $T = N^{\mathsf{t}}\Delta^{\mathsf{t}}_{\mathsf{min}}$ (seconds) and the bandwidth of $B^{\mathsf{f}} = M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}$ (Hz). Considering the scheduling problem for users where the serving base station supports multiple numerologies as shown in Fig 7.1. Particularly, the bandwidth of a PRB in numerology l is equal to one half of that in numerology $l+1$ while the time slot duration of a PRB in numerology l is twice of that in numerology $l+1$. The bandwidth of a PRB in numerology l is defined as Δ^{f}_l and the slot duration of a PRB in numerology l is defined as Δ^{t}_l. Then, we have $\Delta^{\mathsf{t}}_l = \Delta^{\mathsf{t}}_{l-1}/2$, $\Delta^{\mathsf{f}}_l = 2\Delta^{\mathsf{f}}_{l-1}$, $\Delta^{\mathsf{t}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{t}}_l, \forall l\}$, and $\Delta^{\mathsf{f}}_{\mathsf{min}} = \min\{\Delta^{\mathsf{f}}_l, \forall l\}$. For convenience, the numerology used by user k is denoted as l_k, the set of all users is denoted as $\mathcal{K}$, the set of numerologies is denoted as $\mathcal{L}$, $l_{\mathsf{max}} = \max\{l \in \mathcal{L}\}$, $l_{\mathsf{min}} = \min\{l \in \mathcal{L}\}$, and $L = l_{\mathsf{max}} - l_{\mathsf{min}}$, and the cardinals of set $\mathcal{L}$ is denoted as $|\mathcal{L}|$.

Each user k requires a data chunk of d^{rq}_k bits be completely transmitted and the total waiting time for its data transmission must not be larger than τ^{max}_k. Note that d^{rq}_k can be the whole data carried by a data flow (e.g., sensing data) or a part of the data of the underlying data flow (e.g., streaming data) in T seconds.[1] We use (i,j) to refer to a particular RE where its location is given as $f \in [(i-1)\Delta^{\mathsf{f}}_{\mathsf{min}} : i\Delta^{\mathsf{f}}_{\mathsf{min}})$ and $t \in [(j-1)\Delta^{\mathsf{t}}_{\mathsf{min}} : j\Delta^{\mathsf{t}}_{\mathsf{min}})$, for $1 \le i \le M^{\mathsf{f}}$ and $1 \le j \le N^{\mathsf{t}}$.

[1] We assume that the d^{rq}_k is known or can be estimated from the QoS requirement and data properties.

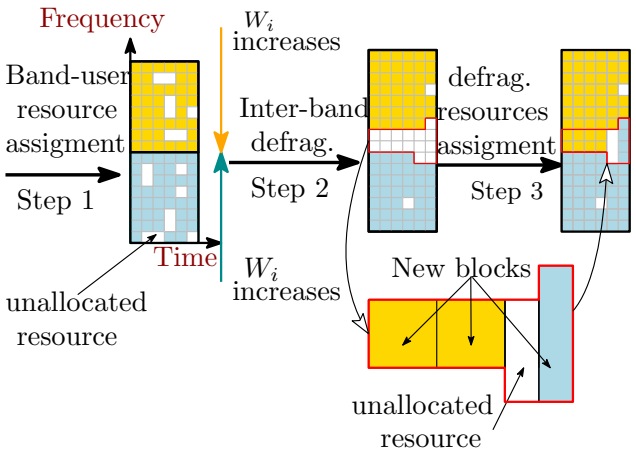

Figure 7.2 – Illustration of the three steps of RPA algorithm

7.2.1 Problem Formulation

PRBs are allocated to users where the numerology is selected in advance by each user. Moreover, we consider the non-overlapping resource allocation design, thus 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:

$$\boldsymbol{q}_{l,m,n} = \{(i,j)|m \leq i \leq m + M_l, n \leq j \leq n + N_l\}, \tag{7.1}$$

where $M_l = 2^{l-l_{\min}}-1$, $N_l = 2^{l_{\max}-l}-1$. 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 $x_{i,j}^{k,c}$'s, $y_{m,n}^{k,c}$'s where $y_{m,n}^{k,c} = 1$ if $\boldsymbol{q}_{l_k,m,n}$ corresponds to the c^{th} assigned PRB of user k, and $y_{m,n}^{k,c} = 0$ otherwise; $x_{i,j}^{k,c} = 1$ if RE (i,j) is assigned for user k in its c^{th} PRB, and $x_{i,j}^{k,c} = 0$ otherwise. For $k \in \mathcal{K}$, the ranges of c, i, j, m, and n are $c = 1 : C_k$, $i = 1 : M^{\mathsf{f}}$, $j = 1 : N^{\mathsf{t}}$, $m = 1 : M^{\mathsf{f}} - M_{l_k}$, and $n = 1 : N^{\mathsf{t}} - N_{l_k}$, respectively. We impose the following constraints to ensure non-overlapping RA:

$$\sum_{i'=m:m+M_{l_k}} \sum_{j'=n:n+N_{l_k}} x_{i',j'}^{k,c} \geq 2^L y_{m,n}^{k,c}, \ \forall k, c, m, n, \tag{7.2a}$$

$$\sum\nolimits_{k\in\mathcal{K}} \sum\nolimits_{c} x_{i,j}^{k,c} \leq 1, \forall i, j, \text{ and } \sum\nolimits_{m} \sum\nolimits_{n} y_{m,n}^{k,c} \leq 1, \forall k, c. \tag{7.2b}$$

Specifically, (7.2a) implies that the number of REs per PRB remains constant and equal to 2^L. Moreover, (7.2b) indicates that each RE should belong to only one PRB, and each PRB can be assigned to at most one user. Let $r_{m,n}^k$ denote the transmission rate of user k on PRB $\boldsymbol{q}_{l_k,m,n}$. Then, the total amount of data transmitted by user k during the scheduling interval is $d_k = \Delta_{l_k}^{\mathsf{t}} \sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}} \sum_{n=1}^{N^{\mathsf{t}}-N_{l_k}} \sum_{c=1}^{C_k} r_{m,n}^k y_{m,n}^{k,c}$.

Each user k wants its data chunk to be completely transmitted and the total waiting time be not larger than τ_k^{max}. Let $\tau_{k,0}$ denote the initial waiting time (of the data chunk) of user $k \in \mathcal{K}$ at the beginning of the considered scheduling interval.[2] 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 $\tau_{k,1} = \Delta_{\mathsf{min}}^{\mathsf{t}} \min\{j-1 \mid x_{i,j}^{k,c}=1, \forall i,j,c\}$.

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^{rq} bits with the total waiting time not larger than τ_k^{max}. To maintain these constraints, we define a function capturing if both constraints are satisfied as $u_k = \mathbb{1}_{d_k - d_k^{\mathsf{rq}}} \mathbb{1}_{\tau_k^{\mathsf{max}} - \tau_k}$, 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 $d_k \geq d_k^{\mathsf{rq}}$ and $\tau_k \leq \tau_k^{\mathsf{max}}$, respectively, we have $u_k = 1$; otherwise, $u_k = 0$. Then, the scheduling problem can be formulated as

$$(\mathcal{P}_1) \max_{\boldsymbol{x},\boldsymbol{y}} \sum\nolimits_{k\in\mathcal{K}} u_k \text{ s.t. } (7.2a), (7.2b), \text{ and } \boldsymbol{x}, \boldsymbol{y} \in \{0,1\},$$

where $\boldsymbol{x} = \{x_{i,j}^{k,c} | \ \forall i,j,k,c\}$ and $\boldsymbol{y} = \{y_{m,n}^{k,c} | \ \forall k,c,m,n\}$.

7.2.2 Problem Transformation

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

$$u_k \in \{0,1\}; \ u_k(\tau_k^{\mathsf{max}} - \tau_k) \geq 0; \ u_k(d_k - d_k^{\mathsf{rq}}) \geq 0. \tag{7.3}$$

Then, we introduce auxiliary variables $z_{m,n}^{k,c}$'s as $z_{m,n}^{k,c} = u_k y_{m,n}^{k,c}$. Using $z_{m,n}^{k,c}$'s, $(\mathcal{P}_1)$ can be transformed into the ILP form as stated in the following proposition.

[2]This initial waiting time is applied to the first chunk of a new data flow when the data flow arrives in the middle of the previous scheduling interval.

Proposition 7.1. *$(\mathcal{P}_1)$ is equivalent to the following problem*

$$
\begin{aligned}
(\mathcal{P}_1^{\mathsf{ILP}}) \quad & \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \ \sum_{k\in\mathcal{K}} u_k \\
& \textit{s. t.} \quad (7.2a),\ (7.2b), \\
& \qquad \sum_{m} \sum_{n'=1:N_k^{\mathsf{rq}}} \sum_{c} z_{m,n'}^{k,c} - u_k \geq 0,\ \forall k \in \mathcal{K}, && \text{(7.4a)} \\
& \qquad \Delta_{l_k}^{\mathsf{t}} \sum_{m}\sum_{n}\sum_{c} r_{m,n}^{k} z_{m,n}^{k,c} - u_k d_k^{\mathsf{rq}} \geq 0,\ \forall k \in \mathcal{K}, && \text{(7.4b)} \\
& \qquad z_{m,n}^{k,c} \geq u_k + y_{m,n}^{k,c} - 1,\ \ z_{m,n}^{k,c} \leq \min\{u_k, y_{m,n}^{k,c}\},\ \forall k,c,m,n, && \text{(7.4c)}
\end{aligned}
$$

where $\boldsymbol{z} = \{z_{m,n}^{k,c}, z_{m,n}^{k,c} |\ \forall k,c,m,n\}$ and $\boldsymbol{u} = \{u_k^{\mathsf{r}}, u_k^{\mathsf{d}} |\ \forall k\}$. Here, N_k^{rq} represents the maximum number of time slots (with size of $\Delta_{\mathsf{min}}^{\mathsf{t}}$ seconds) that user k can wait, counting from the beginning of the scheduling interval, to meet its delay constraint which is determined as $N_k^{\mathsf{rq}} = \lfloor (\tau_k^{\mathsf{max}} - \tau_{k_0})/\Delta_{\mathsf{min}}^{\mathsf{t}} \rfloor$, where $\lfloor . \rfloor$ is the floor operation.

Proof. (7.4c) with $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u} \in \{0,1\}$ are equivalent to $z_{m,n}^{k,c} = u_k y_{m,n}^{k,c}, \forall k,c,m,n$. As a result, (7.4a) and (7.4b) in $(\mathcal{P}_1)$ are equivalent to (7.3). Hence, $(\mathcal{P}_1)$ is equivalent to $(\mathcal{P}_1^{\mathsf{ILP}})$. □

Proposition 7.2. *$(\mathcal{P}_1^{\mathsf{ILP}})$ 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 NP-hard [138]. Thus, $(\mathcal{P}_1^{\mathsf{ILP}})$ is more complex than the standard CSP since the numbers of PRBs assigned for different users should also be optimized. Therefore, $(\mathcal{P}_1^{\mathsf{ILP}})$ must be NP-hard. □

7.3 Proposed Algorithms

7.3.1 Resource Partitioning Based Algorithm (RPA)

We propose a low-complexity algorithm which solves $(\mathcal{P}_1^{\mathsf{ILP}})$ by decomposing it into parallel small-scale sub-problems. 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

Algorithm 7.1. Resource Partitioning based Algorithm (RPA)

1: Initialize: Set initial value for M_{B}.
2: Partition resources into M_{B} sub-bands and distribute users into these sub-bands as in **Section 7.3.1**.
3: **Step 1**: Solve $(\mathcal{P}^{\mathsf{ILP}}_{1,m_{\mathsf{B}}})$ to obtain $u^{\mathsf{S1}\star}_k$ for all m_{B}, $k \in \mathcal{K}_{m_{\mathsf{B}}}$.
4: **Step 2**: Solve $(\mathcal{P}_{m_{\mathsf{B}}})$ to create a contiguous region of unallocated resources between two consecutive sub-bands while still satisfying the requirements of admitted users, i.e., users with $u^{\mathsf{S1}\star}_k = 1$, $\forall k$.
5: **Step 3**: Solve $(\mathcal{P}_{\mathsf{RPA}})$ to assign unallocated resources to un-admitted users, i.e., users with $u^{\mathsf{S1}\star}_k = 0$, $\forall k$.

Algorithm 7.2. Iterative Greedy Algorithm (IGA)

1: Initialize: $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_k, c_k = 0, \mathcal{W}_{m,n} = 1, W = 10$.
2: **repeat**
3: Compute $\mathcal{U}_{m,n,k}$ find the largest value of $\mathcal{U}_{m_0,n_0,k_0}$, and perform the corresponding PRB allocation.
4: Update different parameters after the PRB allocation as $c_{k_0} = c_{k_0} + 1$, assign $y^{k_0,c_{k_0}}_{m_0,n_0} = 1$, update the remaining required data $d^{\mathsf{rq}}_{k,0} = d^{\mathsf{rq}}_{k,0} - r^{k_0}_{m_0,n_0}\Delta^{\mathsf{t}}_{l_k}$, and $\mathcal{W}_{m_0,n} = W, \forall n = 1 : N^{\mathsf{t}}$.
5: Drop all overlapped PRBs $\boldsymbol{q}_{l_k,m,n}$ to PRB $\boldsymbol{q}_{l_{k_0},m_0,n_0}$
6: **until** $\mathcal{U}_{m,n,k} = 0, \forall m, n, k$

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 $(m_{\mathsf{B}} - 1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}}$ to $\min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor\Delta^{\mathsf{f}}_{\mathsf{min}}, M^{\mathsf{f}}\Delta^{\mathsf{f}}_{\mathsf{min}}\}$. Then, the following three steps are taken in RPA: 1) perform RA on each sub-band, 2) re-arrange unallocated resources for consecutive sub-bands, and 3) assign these re-arranged resources to support more (un-admitted) users.

The key steps of RPA are illustrated in Fig. 7.2 and it is summarized in **Algorithm** 7.1. 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_{B} as $\mathcal{K}_{m_{\mathsf{B}}}$, and the set of REs in the frequency dimension as $\mathcal{I}_{m_{\mathsf{B}}} = \{m | m = (m_{\mathsf{B}} - 1)\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor : \min\{m_{\mathsf{B}}\lfloor M^{\mathsf{f}}/M_{\mathsf{B}}\rfloor, M^{\mathsf{f}}\}\}$. We then solve $(\mathcal{P}^{\mathsf{ILP}}_1)$ corresponding to each sub-band m_{B} and the set of users $\mathcal{K}_{m_{\mathsf{B}}}$ to obtain admission decisions, denoted as $u^{\mathsf{S1}\star}_k$'s. The sub-problem for sub-band m_{B} is named $(\mathcal{P}^{\mathsf{ILP}}_{1,m_{\mathsf{B}}})$. In **Step 2**, after finding $u^{\mathsf{S1}\star}_k$'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 (7.1). The re-arrangement of unallocated REs on

sub-band m_{B} can be achieved by solving the following problem:

$$(\mathcal{P}_{m_{\mathsf{B}}}) \quad \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in\{0,1\}} \sum_{k\in\mathcal{K}_{m_{\mathsf{B}}}} \sum_{i\in\mathcal{I}_{m_{\mathsf{B}}}} \sum_{j=1:N^{\mathsf{t}}-N_{l_k}} \sum_{c=1:C_k} W_i x_{i,j}^{k,c}$$
$$\text{s.t.} \quad u_k = u_k^{\mathsf{S1}\star}, \quad (7.2a), (7.2b), (7.4a), (7.4b), \ \forall k\in\mathcal{K}_{m_{\mathsf{B}}}, i\in\mathcal{I}_{m_{\mathsf{B}}},$$

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 $(\mathcal{P}_{m_{\mathsf{B}}})$, 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 $\{\boldsymbol{x}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}, \boldsymbol{y}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}, \boldsymbol{z}^\star_{\mathcal{P}_{m_{\mathsf{B}}}}\}$ denote the optimal solution of $(\mathcal{P}_{m_{\mathsf{B}}})$. In **Step 3**, we assign the unallocated resources, $\Omega = \{(i,j)|\boldsymbol{x}^\star_{\mathcal{P}_{m_{\mathsf{B}}}} = 0, \forall m_{\mathsf{B}}\}$, to the set of unadmitted users $\bar{\mathcal{K}} = \{k|u_k^{\mathsf{S1}\star} = 0\}$ by solving the following problem $(\mathcal{P}_{\mathsf{RPA}})$:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u}\in\{0,1\}} \sum_{k\in\bar{\mathcal{K}}} u_k \text{ s.t. } (7.2a), (7.2b), (7.4a), (7.4b), \forall k\in\bar{\mathcal{K}}, (i,j)\in\Omega.$$

7.3.2 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: $\mathcal{U}_{m,n}^k = \frac{r_{m,n}^k \Delta_{l_k}^{\mathsf{t}}}{d_{k,0}^{\mathsf{rq}}} \frac{n}{N_k^{\mathsf{rq}}} \mathbb{1}_{n\in\mathcal{N}_k} \mathcal{W}_{m,n}$ if $c_k \le C_k, d_{k,0}^{\mathsf{rq}} > 0$, and $\mathcal{U}_{m,n}^k = 0$, otherwise, where $d_{k,0}^{\mathsf{rq}}$ is the remaining required data amount in each iteration, which is equal to d_k^{rq} in the first iteration, c_k is the current total PRBs assigned to user k, and $\mathcal{N}_k$ is the set of REs in the time domain, which is defined as $\mathcal{N}_k = \{n|n \le N_k^{\mathsf{rq}}\}$ if $\sum_{n=1}^{N_k^{\mathsf{rq}}} \sum_{m=1}^{M^{\mathsf{f}}-M_{l_k}} \sum_{c=1}^{C_k} y_{m,n}^{k,c} = 0$, and $\mathcal{N}_k = \{n|n \le N^{\mathsf{t}}\}$, otherwise, and $\mathcal{W}_{m,n}$ 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 $\mathcal{W}_{m,n}$. In particular, $\mathcal{U}_{m,n}^k$ 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 7.2**. In each iteration of IGA, we need to compute the resource assignment weights $\mathcal{U}_{m,n,k}$ for all available m, n, k, determine the largest $\mathcal{U}_{m_0,n_0,k_0}$ to perform RA, update different parameters, and drop all overlapped PRBs to the assigned block. The worst-case complexity of each iteration is $\mathcal{O}(M^{\mathsf{f}}N^{\mathsf{t}}K)$. Let N^{iter} be the number of iterations, which is upper bounded by $M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|$, the overall worst-case complexity of IGA is $\mathcal{O}(N^{\mathsf{iter}}M^{\mathsf{f}}N^{\mathsf{t}}|\mathcal{L}|K)$.

7.4 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 β_k (dB) $= 128.1+37.6\log_{10}(\gamma_k)$ where γ_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 the same. Specifically, $\lfloor K/3 \rfloor$, $\lfloor K/3 \rfloor$ and $K–2\lfloor K/3 \rfloor$ for group A, B, and C, 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 2.8×10^{11}. The required data chunks d_k^{rq} 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 and uniformly in [3-6] and [1-4] for users in groups B and C, respectively. All numerical results are obtained by averaging the results over 50 random realizations.

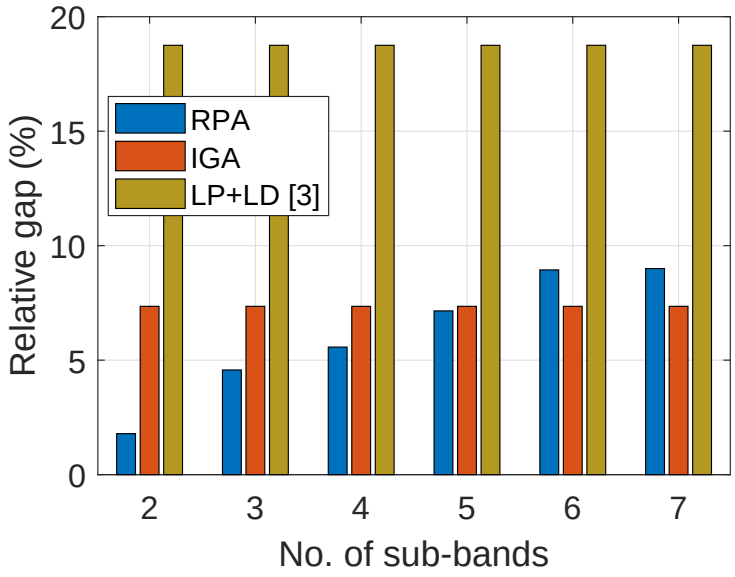

Figure 7.3 – Comparison of RPA and IGA with the optimum on the relative gap

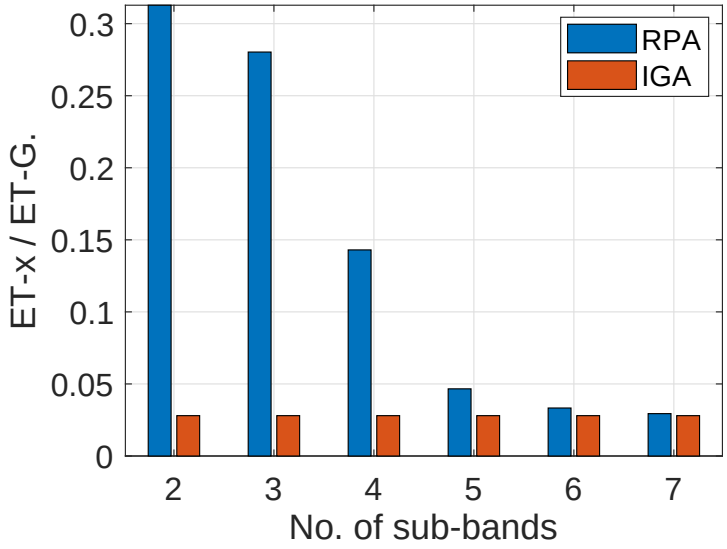

Figure 7.4 – Comparison of RPA and IGA with the optimum on the execution time

We show the relative gap in Fig. 7.3 which is calculated as $(\sum_k u_k^{\mathsf{G}} - \sum_k u_k^{\mathsf{RPA/IGA}}) \times 100\% / \sum_k u_k^{\mathsf{G}}$ for $M^{\mathsf{f}} = 32$ and $K = 20$ where $u_k^{\mathsf{RPA/IGA/LP+LD[3]}}$ and u_k^{G} represent the objective values for user k obtained by using RPA/IGA/LP+LD[4] and the CVX-Gurobi solver, respectively. Fig. 7.4 shows the ratio between the average execution time (ET) of the RPA/IGA and that required to solve $(\mathcal{P}_1^{\mathsf{ILP}})$ by the CVX-Gurobi solver (ET-G). We have chosen u_k^{G} as a referenced solution because the relative gap between the actual optimal solution and the solution returned by the Gurobi solver is less than 10^{-4} by default. The "LP+LD" algorithm proposed in [4] 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. Specifically, the utility matrix is determined by considering the linear programming (LP) relaxation and the Lagrangian dual (LD) problem. In fact, the "LP+LD" algorithm obtains its solution by tackling the ILP dual problem where the zero duality-gap is generally not guaranteed. 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} [3].

Fig. 7.5 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 $M^{\mathsf{f}} = 64$ in Fig. 7.6. 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 $d_k^{\mathsf{rq}} = 250$ 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.

7.5 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. Particularly, strict latency requirements

[3]By implementing the code on Matlab to solve ($\mathcal{P}_1^{\mathsf{ILP}}$) with Gurobi using a computer equipped with CPU chipset Intel core i7-4790 and 12 GB RAM, average execution time is about 6.39 seconds when $M^{\mathsf{f}} = 32$ and $K = 20$.

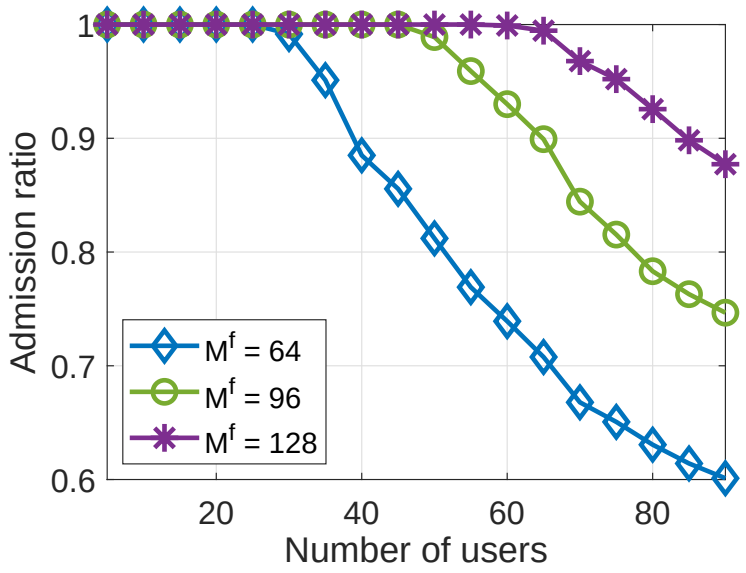

Figure 7.5 – Admission ratio due to IGA for different number of users.

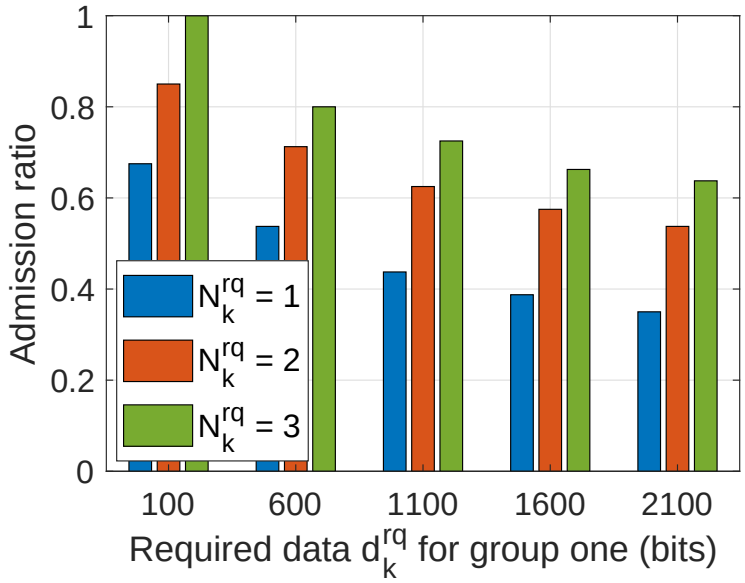

Figure 7.6 – Admission ratio due to IGA for different required data.

can lead to very negative impacts of the admission control performance and choosing a suitable M_{B} in RPA can greatly reduce the computation complexity.

Chapter 8

Conclusions and Further Work

In this chapter, we summarize our research contributions and discuss some potential directions for further research.

8.1 Major Research Contributions

In the first set of contributions [8, 9, 126, 139], we have investigated the energy-efficient resource allocation and computation offloading in MEC systems. Particularly, we formulate the computation task offloading and resource allocation optimization in MEC systems considering different advanced wireless technologies including device-to-device (D2D) [8], heterogeneous network (HetNet) [9], unmanned aerial vehicle (UAV) [139], and multiple-input multiple output (MIMO) communications [126]. The minimization of total energy consumption is considered in [8, 139], and the fairness on resource allocation for energy saving is presented in [9, 126]. We then propose different algorithms to tackle these underlying problems. The proposed algorithms are shown to perform significantly better than conventional local computation strategies via numerical studies..

In the second set of contributions [140], we propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by data compression and decompression. We then formulate the joint optimization of the compression ratio, computation offloading, and resource allocation to minimize the maximum weighted energy and service delay cost (WEDC) of all users considering data compression at only the mobile users and at both the mobile users

and the fog server. First, when data compression is performed only at the mobile users, we prove that the optimal offloading decisions have a threshold structure. Moreover, a novel three-step approach employing convexification techniques is developed to optimize the compression ratios and the resource allocation. Then, we address the more general design where data compression is performed at both the mobile users and the fog server. We propose three algorithms to overcome the strong coupling between the offloading decisions and the resource allocation. We show that our proposed designs outperform conventional computation offloading strategies that do not leverage data compression or use sub-optimal optimization approaches.

In the final set of contributions [141, 142], 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. We propose to transform the underlying IP scheduling problem into an ILP and then present two low-complexity algorithms, named RPA and IGA, to acquire efficient resource scheduling solutions. The RPA algorithm is developed by partitioning the resources and users into smaller groups, optimizing the RA for each group, then performing resource defragmentation and additional resource assignment for unallocated resources to obtain a final solution. In the IGA algorithm, we iteratively allocate PRBs to users based on an appropriate resource assignment weight to obtain an efficient scheduling solution with low computation complexity. Finally, extensive numerical studies are performed to demonstrate the efficacy of the proposed algorithms.

8.2 Further Research Directions

The following research directions are of importance and deserve further investigation.

8.2.1 Dense MEC Systems

For the research study conducted in this dissertation, we have only considered systems with one edge server. However, when the number of mobile devices becomes sufficiently large, a single edge server may not provide enough the computing resource to support all users. Therefore, dense MEC systems with a large number of deployed edge servers can allow to cope with large users' computation demand. However, one must address further research challenges for such systems

including interference management in dense wireless networks as well as joint computation load balancing and resource allocation.

8.2.2 UAV Based MEC Systems

It has been recently recognized that the UAV based MEC system can be employed to support highly mobile users or certain application scenarios [68–75]. Mobile cloudlet-mounted UAVs can be deployed as a mobile MEC system to support special communications scenarios such as temporary events and they can reconfigured to adaptively meet certain users' demand and behaviors. In particular, the path planning is an important design issue of the mobile MEC system. Moreover, joint optimization of UAVs' path planning, computing, and radio resource allocation is a challenging research problem deserving more research.

8.2.3 Machine Learning Applications for 5G NR and MEC Systems

In the wireless environment, the time-varying wireless channel strongly impacts the involved decision making including resource allocation, wireless scheduling in 5G NR and MEC systems. Developing online algorithms that can adapt to the time-varying wireless environment could allow us to better manage the resource and network and achieve the best network performance. Deep reinforcement learning and deep learning are powerful design tools to achieve these goals. Hence, applications of deep reinforcement learning and deep learning to engineer efficient resource management algorithms in 5G NR and MEC systems present an interesting research direction for our future research.

8.3 List of Publications

8.3.1 Journals

[J4]. Ti Ti Nguyen, Long B. Le, and Quan Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. Serv. Comput.,* Early Access, Jan. 2019.

[J3]. Ti Ti Nguyen, Vu Nguyen Ha, Long B. Le, and Robert Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," *IEEE Trans. Wireless Commun.,* vol. 19, no. 1, pp. 293–309, Jan. 2020.

[J2]. Ti Ti Nguyen, Vu Nguyen Ha, and Long Bao Le, "Wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks," *IEEE Commun. Letters,* vol. 24, no. 2, pp. 410-413, Feb. 2020.

[J1]. Vu Nguyen Ha, Ti Ti Nguyen, Long B. Le, and Frigon, Jean-François, "Admission control and network slicing for multi-numerology 5G wireless networks," *IEEE Netw. Letters,* vol. 19, no. 1, pp. 293–309, Jan. 2020.

8.3.2 Conferences

[C4]. Ti Ti Nguyen and Long Bao Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017.

[C3]. Ti Ti Nguyen and Long Bao Le, "Joint computation offloading and resource allocation in cloud based wireless HetNets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017.

[C2]. Ti Ti Nguyen and Long Bao Le, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018.

[C1]. Ti Ti Nguyen and Long Bao Le, "Joint resource allocation, computation offloading, and path planning for UAV based hierarchical fog-cloud mobile systems," in *Proc. IEEE Int. Conf. Commun. Elec. (ICCE)*, July 2018.

Références

[1] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[2] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025". [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[3] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, Sept. 2018.

[4] L. You, Q. Liao, N. Pappas, and D. Yuan, "Resource optimization with flexible numerology and frame structure for heterogeneous services," *IEEE Commun. Letters*, vol. 22, no. 12, pp. 2579–2582, Aug. 2018.

[5] A. Akhtar and H. Arslan, "Downlink resource allocation and packet scheduling in multi-numerology wireless systems," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2018, pp. 362–367.

[6] S. Lagen, B. Bojovic, S. Goyal, L. Giupponi, and J. Mangues-Bafalluy, "Subband configuration optimization for multiplexing of numerologies in 5G TDD new radio," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, 2018, pp. 1–7.

[7] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[8] T. T. Nguyen and L. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[9] ——, "Joint computation offloading and resource allocation in cloud based wireless HetNets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2017, pp. 1–6.

[10] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, July 2017.

[11] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[12] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[13] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[14] H. Shah-Mansouri and V. W. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246–3257, Dec. 2018.

[15] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[16] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[17] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, Apr. 2017.

[18] K. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Energy efficiency in massive MIMO-based 5G networks: Opportunities and challenges," *IEEE wireless commun.*, vol. 24, no. 3, pp. 86–94, Jan. 2017.

[19] N. Xia, H.-H. Chen, and C.-S. Yang, "Radio resource management in machine-to-machine communications-a survey," *IEEE Commun. Surveys Tutorials*, Oct. 2017.

[20] E. Standards. Multi-access Edge Computing (MEC). [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[21] S. Peter, K. Benedek, T. Stephen, and W. Peter, "Next-generation edge-cloud ecosystem," *Ericsson Tech. Review*, pp. 1–11, 2020.

[22] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[23] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. Da Silva, "Waveform and numerology to support 5G services and requirements," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 90–98, Nov. 2016.

[24] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology.* Academic Press, 2018.

[25] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[26] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[27] 3GPP-TR-36.814, "Evolved universal terrestrial radio access (E-UTRA); further advancements for E-UTRA physical layer aspects (release 9)," Tech. Rep., 2010.

[28] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sept. 2013.

[29] S. Martello, *Knapsack problems: algorithms and computer implementations.* John Wiley & Sons Ltd., 1990.

[30] M. Sanjabi, M. Razaviyayn, and Z.-Q. Luo, "Optimal joint base station assignment and beamforming for heterogeneous networks." *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1950–1961, Apr. 2014.

[31] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[32] Y. Yu, B. Krishnamachari, and V. P. Kumar, *Information processing and routing in wireless sensor networks.* World Scientific, 2006.

[33] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, Nov. 2017.

[34] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397 – 1411, Mar. 2017.

[35] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[36] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11 255–11 268, Jun. 2017.

[37] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Nov. 2018.

[38] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, June 2019.

[39] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[40] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Apr. 2018.

[41] J. Ren, G. Yu, Y. He, and Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[42] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, June 2012.

[43] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan. 2015.

[44] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Commun. Letters*, vol. 3, no. 2, pp. 153–156, Apr. 2014.

[45] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[46] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[47] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, Aug. 2016.

[48] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sept. 2017.

[49] Y. Geng, Y. Yang, and G. Cao, "Energy-efficient computation offloading for multicore-based mobile devices," in *Proc. IEEE INFOCOM*, 2018.

[50] M.-H. Chen, M. Dong, and B. Liang, "Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, Dec. 2018.

[51] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 313 – 12 325, Dec. 2018.

[52] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Mar. 2017.

[53] M. Liu, Y. Mao, S. Leng, and S. Mao, "Full-duplex aided user virtualization for mobile edge computing in 5G networks," *IEEE Access*, vol. 6, pp. 2996–3007, Dec. 2017.

[54] L. Ji and S. Guo, "Energy-efficient cooperative resource allocation in wireless powered mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4744–4754, Nov. 2018.

[55] Z. Zhu, J. Peng, X. Gu, H. Li, K. Liu, Z. Zhou, and W. Liu, "Fair resource allocation for system throughput maximization in mobile edge computing," *IEEE Access*, vol. 6, pp. 5332–5340, Jan. 2018.

[56] J. Zhou, X. Zhang, and W. Wang, "Joint resource allocation and user association for heterogeneous services in multi-access edge computing networks," *IEEE Access*, vol. 7, pp. 12 272–12 282, jan. 2019.

[57] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Feb. 2019.

[58] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4642–4655, June 2018.

[59] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet .Things J.*, vol. 5, no. 4, pp. 2633–2645, Dec. 2017.

[60] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jan. 2018.

[61] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, jan. 2018.

[62] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, Feb. 2019.

[63] F. Wang, J. Xu, and Z. Ding, "Multi-antenna noma for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Nov. 2018.

[64] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and resource allocation for D2D-enabled mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Mar. 2019.

[65] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 169–11 185, Aug. 2019.

[66] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Dec. 2018.

[67] C. Zhao, Y. Cai, A. Liu, M. Zhao, and L. Hanzo, "Mobile edge computing meets mmwave communications: Joint beamforming and resource allocation for system delay minimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, Apr. 2020.

[68] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in uav-enabled mobile edge computing networks with multiple service providers," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8753–8769, June 2019.

[69] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3688–3699, Dec. 2018.

[70] J. Zhang, L. Zhou, F. Zhou, B.-C. Seet, H. Zhang, Z. Cai, and J. Wei, "Computation-efficient offloading and trajectory scheduling for multi-uav assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, Dec. 2019.

[71] J. Hu, M. Jiang, Q. Zhang, Q. Li, and J. Qin, "Joint optimization of uav position, time slot allocation, and computation task partition in multiuser aerial mobile-edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7231–7235, May 2019.

[72] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in uav-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, July 2019.

[73] X. Zhang, Y. Zhong, P. Liu, F. Zhou, and Y. Wang, "Resource allocation for a uav-enabled mobile-edge computing system: Computation efficiency maximization," *IEEE Access*, vol. 7, pp. 113 345–113 354, Aug. 2019.

[74] X. Diao, J. Zheng, Y. Wu, Y. Cai, and A. Anpalagan, "Joint trajectory design, task data, and computing resource allocations for noma-based and uav-assisted mobile edge computing," *IEEE Access*, vol. 7, pp. 117 448–117 459, Aug. 2019.

[75] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3 424–3 438, Jan. 2020.

[76] J. Yan, S. Bi, Y.-J. A. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 235–250, Jan. 2020.

[77] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, 2019, Early Access.

[78] H. Ke, J. Wang, H. Wang, and Y. Ge, "Joint optimization of data offloading and resource allocation with renewable energy aware for iot devices: A deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 179 349–179 363, Dec. 2019.

[79] L. Lei, H. Xu, X. Xiong, K. Zheng, W. Xiang, and X. Wang, "Multiuser resource control with deep reinforcement learning in iot edge computing," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 119–10 133, Aug. 2019.

[80] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 158–11 168, Aug. 2019.

[81] R. Abreu, T. Jacobsen, K. Pedersen, G. Berardinelli, and P. Mogensen, "System level analysis of eMBB and grant-free URLLC multiplexing in uplink," in *IEEE Veh. Technol. Conf. (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[82] E. Fountoulakis, N. Pappas, Q. Liao, V. Suryaprakash, and D. Yuan, "An examination of the benefits of scalable TTI for heterogeneous traffic management in 5G networks," in *Proc. IEEE Int. Symp. Model. Opt. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2017, pp. 1–6.

[83] Q. Liao, P. Baracca, D. Lopez-Perez, and L. G. Giordano, "Resource scheduling for mixed traffic types with scalable TTI in dynamic TDD systems," in *Proc. IEEE Global Commun. Conf. Workshops (GLOBECOM Wkshps)*, 2016, pp. 1–7.

[84] R. Kassab, O. Simeone, and P. Popovski, "Coexistence of URLLC and eMBB services in the C-RAN uplink: an information-theoretic study," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[85] A. Yazar and H. Arslan, "A flexibility metric and optimization methods for mixed numerologies in 5G and beyond," *IEEE Access*, vol. 6, pp. 3755–3764, 2018.

[86] L. Marijanovic, S. Schwarz, and M. Rupp, "Optimal numerology in OFDM systems based on imperfect channel knowledge," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, 2018, pp. 1–5.

[87] J. Choi, B. Kim, K. Lee, and D. Hong, "A transceiver design for spectrum sharing in mixed numerology environments," *IEEE Trans. Wireless Commun.*, vol. 18, no. 5, pp. 2707–2721, Apr. 2019.

[88] R. Kassab, O. Simeone, P. Popovski, and T. Islam, "Non-orthogonal multiplexing of ultra-reliable and broadband services in fog-radio architectures," *IEEE Access*, vol. 7, pp. 13 035–13 049, Jan. 2019.

[89] J. Tang, B. Shim, and T. Q. Quek, "Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 04, pp. 881–895, Feb. 2019.

[90] M. Chiang, C. W. Tan, D. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, July 2007.

[91] H. Tuy, "Global minimization of a difference of two convex functions," in *Nonlinear Analysis and Optimization*. Springer, 1987, pp. 150–182.

[92] C. Rudin, "Lecture notes in prediction: Machine learning and statistics," May 2012.

[93] H. H. Kha, H. D. Tuan, and H. H. Nguyen, "Fast global optimal power allocation in wireless networks by local dc programming," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 510–515, Feb. 2012.

[94] M. Othman, S. A. Madani, S. U. Khan *et al.*, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, July 2013.

[95] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM MobiSys*, 2010, pp. 49–62.

[96] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. ACM EuroSys*, 2011, pp. 301–314.

[97] S. Kosta, A. Aucinas, H. Pan, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, 2012, pp. 945–953.

[98] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Aug. 2013.

[99] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. Hot Topics Cloud Compt. (HotCloud)*, 2010, pp. 4–11.

[100] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, 2015.

[101] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NIPS*, 2014, pp. 568–576.

[102] S. E. Mahmoodi, R. Uma, and K. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comp.*, vol. PP, no. 99, Apr. 2016.

[103] A. A. Zaidi, R. Baldemair, M. Andersson, S. Faxér, V. Molés-Cases, and Z. Wang, "Designing for the future: the 5G NR physical layer," *Ericsson Tech. Review*, pp. 1–13, 2017.

[104] F. Rayal, "Lte in a nutshell: The physical layer," *Telesystem Innovations*, 2010.

[105] J. Campos, "Understanding the 5G NR physical layer," Keysight Technologies, Tech. Rep., 2017.

[106] A. B. Kihero, M. S. J. Solaija, and H. Arslan, "Inter-numerology interference for beyond 5G," *IEEE Access*, vol. 7, pp. 146 512–146 523, 2019.

[107] Q. Technologies, "Making 5G NR a reality," Qualcomm Technologies, Tech. Rep., 2016.

[108] D. Kong, "Science driven innovations powering mobile product: Cloud AI vs. device AI solutions on smart device," *arXiv preprint arXiv:1711.07580*, 2017.

[109] S. Rallapalli, H. Qiu, A. Bency, S. Karthikeyan, R. Govindan, B. Manjunath, and R. Urgaonkar, "Are very deep neural networks feasible on mobile devices," in *Proc. ACM HotMobile Workshop*, 2016.

[110] R. Van Noorden, "A better battery," *Nature News*, vol. 507, no. 7490, pp. 26–28, 2014.

[111] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Sig. Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[112] H. Flores and S. N. Srirama, "Mobile cloud middleware," *J. Sys. and Soft.*, vol. 92, no. 1, pp. 82–94, June 2014.

[113] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[114] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Trans. Cloud Comp.*, Jan. 2018, Early access.

[115] S. Sardellittia, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Infor. Process. Netw.*, vol. 1, no. 2, pp. 89 – 103, June 2015.

[116] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE Trans. Signal Informat. Process. Netw.*, vol. 3, no. 4, pp. 787–802, Dec. 2017.

[117] E. Hossain, M. Rasti, and L. B. Le, "Radio resource management in wireless networks: An engineering approach," *Cambridge University Press*, 2017.

[118] B. V. Patil, P. S. Nataraj, and S. Bhartiya, "Global optimization of mixed-integer nonlinear (polynomial) programming problems: the Bernstein polynomial approach," *Springer Computing*, vol. 94, no. 2-4, pp. 325–343, Mar. 2012.

[119] A. Nemirovski, "Interior point polynomial time methods in convex programming," *Lecture notes*, 2004.

[120] 3GPP-TR-36.101, "Evolved universal terrestrial radio access (E-UTRA); user equipment (UE) radio transmission and reception (release 13)," Tech. Rep., 2016.

[121] Y. Li, M. Sheng, X. Wang, Y. Zhang, and J. Wen, "Max-min energy-efficient power allocation in interference-limited wireless networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 4321–4326, Sept. 2015.

[122] M. Liu, Y. Mao, and S. Leng, "Cooperative fog-cloud computing enhanced by full-duplex communications," *IEEE Commun. Letters*, vol. 22, no. 10, pp. 2044–2047, Oct. 2018.

[123] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for IoT," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 245–254, Apr. 2017.

[124] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Rate-distortion balanced data compression for wireless sensor networks," *IEEE J. Sensors*, vol. 16, no. 12, pp. 5072–5083, Apr. 2016.

[125] W. Zhang, Y. Wen, Y. J. Zhang, F. Liu, and R. Fan, "Mobile cloud computing with voltage scaling and data compression," in *Proc. IEEE Workshop. SPAWC*, 2017, pp. 1–5.

[126] T. T. Nguyen, L. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. Services Comput.*, 2019, Early access.

[127] M. Powell. The canterbury corpus. [Online]. Available: http://corpus.canterbury.ac.nz/descriptions/#cantrbry

[128] T. C. museum of nature. Wallpaper. [Online]. Available: https://nature.ca/en/explore-nature/blogs-videos-more/wallpaper

[129] T. T. Nguyen, V. N. Ha, L. Le, and R. Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," Tech. Rep., Mar. 2019. [Online]. Available: https://arxiv.org/abs/1903.08566

[130] CloudSigma. Are we stealing from you? Understanding CPU steal time in the cloud. [Online]. Available: https://www.cloudsigma.com/understanding-cpu-steal-time-in-the-cloud/

[131] T. D. Hoang, L. B. Le, and T. Le-Ngoc, "Energy-efficient resource allocation for D2D communications in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 6972–6986, Sept. 2016.

[132] N. Saxena, B. J. Sahu, and Y. S. Han, "Traffic-aware energy optimization in green lte cellular systems," *IEEE Commun. Letters*, vol. 18, no. 1, pp. 38–41, Dec. 2013.

[133] R. Méndez-Rial, C. Rusu, A. Alkhateeb, N. González-Prelcic, and R. W. Heath, "Channel estimation and hybrid combining for mmwave: Phase shifters or switches?" in *Proc. Information Theory and Applications Workshop (ITA)*, 2015, pp. 90–97.

[134] K. Li, "Computation offloading strategy optimization with multiple heterogeneous servers in mobile edge computing," *IEEE Trans. Sustainable Comput.*, 2019, Early access.

[135] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Comm. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[136] 3GPP-TS-38.211, "Physical channel and modulation (release 15)," Tech. Rep., 2019.

[137] K. I. Pedersen, G. Berardinelli, F. Frederiksen, P. Mogensen, and A. Szufarska, "A flexible 5G frame structure design for frequency-division duplex cases," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 53–59, Mar. 2016.

[138] C. McDiarmid, "Pattern minimisation in cutting stock problems," *Discrete applied mathematics*, vol. 98, no. 1-2, pp. 121–130, 1999.

[139] T. T. Nguyen and L. Le, "Joint resource allocation, computation offloading, and path planning for uav based hierarchical fog-cloud mobile systems," in *Proc. IEEE Int. Conf. Commun. Elec. (ICCE)*, 2018, pp. 373–378.

[140] T. T. Nguyen, V. N. Ha, L. B. Le, and R. Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 293–309, Jan. 2020.

[141] T. T. Nguyen, V. N. Ha, and L. B. Le, "Wireless scheduling for heterogeneous services with mixed numerology in 5G wireless networks," *IEEE Commun. Letters*, vol. 24, no. 2, pp. 410–413, Feb. 2020.

[142] V. N. Ha, T. T. Nguyen, L. B. Le, and J.-F. Frigon, "Admission control and network slicing for multi-numerology 5G wireless networks," *IEEE Netw. Letters*, vol. 19, no. 1, pp. 293–309, Jan. 2020.