Efficient Resource Allocation for Relay-Assisted Computation Offloading in Mobile Edge Computing

Xihan Chen, *Student Member, IEEE*, Yunlong Cai, *Senior Member, IEEE*, Qingjiang Shi, *Senior Member, IEEE*,
Min-Jian Zhao, *Member, IEEE*, Benoit Champage, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

Abstract—Relay-assisted computation offloading (RACO) is investigated, where user A wishes to share the results of computational tasks with another user B with the assistance of a mobile edge relay server (MERS). To enable this computation offloading, we propose a hybrid relaying (HR) approach employing a pair of orthogonal frequency bands, which are respectively used for the amplify-forward relaying of computational results and the decode-forward relaying of the unprocessed raw tasks. The motivation here is to adapt the allocation of computing and communication resources both to dynamic user requirements and to diverse computational tasks. Using this framework, we seek to minimize the weighted sum of the execution delays and the energy consumption in the RACO system by jointly optimizing the computation offloading ratio, the bandwidth allocation, the processor speeds, as well as the transmit power levels of both user A and the MERS, under some practical constraints. By adopting a series of transformations, we first recast this problem into a form amenable to optimization and then develop an efficient iterative algorithm for its solution based on the concave-convex procedure (CCCP). By virtue of the particular problem structure in our case, we propose furthermore a simplified algorithm based on the inexact block coordinate descent (IBCD) method, which leads us to much lower computational complexity. Finally, our numerical results demonstrate the advantages of the proposed algorithms over the state-of-the-art benchmark schemes.

Mobile edge computing, computation offloading, hybrid relaying, resource allocation, CCCP, IBCD.

I. Introduction

Owing to the ever-increasing popularity of smart mobile devices, mobile data traffic continues to grow. According to a recent study [1], the global mobile data traffic will grow at a compound annual growth rate of 46 percent from 2017 to 2022, reaching 77.5 exabytes per month by 2022. Meanwhile, the type of wireless services is also experiencing a major change, expanding from the traditional voice, e-mail and web browsing, to sophisticated applications such as augmented reality, face recognition and natural language processing, to name a few. These emerging services are both latency-sensitive and computation-intensive, hence requiring a reliable low-latency air interface and vast computational resources. In effect, the limited computational capability and battery life of mobile devices cannot guarantee the quality of user experience (QoE) expected for these new services.

To alleviate the performance bottleneck, mobile edge computing (MEC), a new network architecture that supports cloud computing along with Internet service at the network edge, is currently the focus of great attention within the telecommunication industry. Due to the proximity of the mobile devices to the MEC server [2], this architecture has the potential to significantly reduce latency, avoid congestion and prolong the battery lifetime of mobile devices by running demanding applications and processing tasks at the network edge, where ample computational and storage resources remain available [3]. Recently, MEC has gained considerable interest within the research community [4]–[14]. In [4] and [5], the authors derived the optimal resource allocation solution for a single-user MEC system engaged in multiple elastic tasks, aiming to minimize the average execution latency of all tasks under power constraints. A multi-user MEC system was considered in [6], where a game-theoretic model is employed to design computation offloading algorithms for both energy and latency minimization at mobiles. In [7], You *et al.* investigated the optimal resource and offloading decision policy for minimizing the weighted sum of mobile energy consumption under computation latency constraint in a multiuser MEC system based on either time division multiple access (TDMA) or orthogonal frequency division multiple access (OFDMA). Different from the deterministic task model considered in the above works, resource allocation strategies have also been developed for multi-user MEC systems under the stochastic task model, which is characterized by random task arrivals [8]–[10]. For a multi-cell MEC system, the resource management strategies for system performance improvement are more sophisticated. Sardellitti *et al.* [11] considered the joint optimization of radio and computational resources for computation offloading in a dense deployment scenario, in the presence of intercell interference. To overcome the performance bottleneck caused by the extremely high channel state information (CSI) signaling overhead in the centralized algorithm, Wang *et al.* [12] pre-

X. Chen, Y. Cai and M. Zhao are with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China (e-mail: chenxihan@zju.edu.cn; ylcai@zju.edu.cn; mjzhao@zju.edu.cn).
Q. Shi is with the School of Software Engineering, Tongji University, Shanghai, China (e-mail: qing.j.shi@gmail.com).
B. Champagne is with the Department of ECE, McGill University, Montreal, Canada (e-mail: benoit.champagne@mcgill.ca).
L. Hanzo is with the Department of ECS, University of Southampton, Southampton, U.K. (e-mail: lh@ecs.soton.ac.uk).
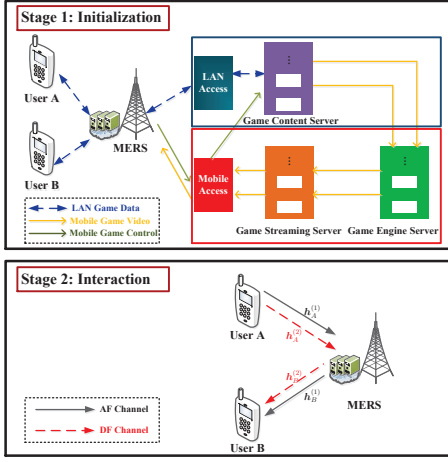
Fig. 1: Illustration of a delay-sensitive mobile game application under the proposed RACO system with HR architecture.

sented a decentralized algorithm based on the alternating direction method of multipliers (ADMM) for joint computation offloading, resource allocation and Internet content caching optimization in heterogeneous wireless networks with MEC. Research efforts have been devoted to the hardware design of MEC systems. For instance, Wang et al. [13] investigated partial computation offloading using both dynamic voltage and frequency scaling[1]. Barbarossa et al. [14] amalgamated MEC-based computation offloading techniques with millimeter wave (mmWave) communications and tackled the intermittency of mmWave links by relying on multiple links.

The aforementioned studies focus on a common scenario, where mobile terminals first offload their computational tasks to the MEC server, which then feeds back the results to the mobile terminals. In contrast to prior studies, we consider a *relay-assisted computation offloading* (RACO) scenario, where user A aims to share its computational results with another user B through a relay platform equipped with MEC capabilities, referred to as a mobile edge relay server (MERS). The underlying ideas behind exploiting the relaying techniques in the RACO system are elaborated on below. From the perspective of communications, the relaying techniques can be considered as cost-effective and promising solutions to increase the throughput, robustness and limited communication range of the RACO systems, due to their simplicity and intuitive designs [15]. In addition, the MERS is equipped with ample computational and storage resources to support computation-intensive and latency-sensitive applications, hence freeing the resource-limited mobile devices from these tasks.

Consequently, a wide variety of applications such as virtual reality (VR) gaming [16], augmented reality (AR) [17], and social networking [18], can benefit from the RACO services. Specifically, the availability of reliable low-latency interactive streaming between two or more users is crucial for supporting these emerging applications, which impose very stringent

requirements on both the computational and radio resources. For clarity, a toy example of specific VR gaming application under the proposed RACO architecture is shown in Fig. 1 to illustrate the practicality of the scenario, where the entire process is divided into two phases, i.e., initialization and interaction. In the initialization phase, user A and B first access the MERS, and the requested connection is confirmed by the game content server [16]. A game engine server is initialized by loading all users' account information and game data from the game content server. Then the game logic and user data are processed to render the raw game video [19]. Finally, a game streaming server is activated to encode the generated raw game video, and the results are sent to each user via wireless links [20]. In the interactive phase, user A and user B take some actions aiming to win the mobile game, where each action corresponds to a specific computational task. Moreover, the result of a specific action taken by any user is bound to significantly affect the final outcome of the mobile game. Hence, to actively participate in the game, each user takes the corresponding action as a counterattack until finding out, which specific action the opponent has taken.

For this type of scenarios, user A can only perform a fraction of its tasks locally due to hardware limitations while the remaining fraction is transferred to the MERS, where more extensive resources are available. This example strongly motivates the need for an efficient and flexible relaying schemes to support computation offloading in MEC systems. Unfortunately, the existing relaying schemes [15] proposed for conventional wireless networks cannot be directly applied to MEC, due to the following reasons. First, different from the conventional wireless networks, the overall performance of MEC is substantially affected by both communication and computational aspects; hence, a novel design criterion that embraces both aspects should be considered. Second, there is a wide variety of emerging applications that could benefit from RACO, but for which the choice of relaying scheme, i.e., amplify-and-forward (AF) versus decode-and-forward (DF), may have a major impact on performance and quality of experience. Motivated by the above considerations, we propose a novel hybrid relay (HR) architecture for the RACO system for improving the exchange of computational results between different users. In this setup, a fraction of the available bandwidth is assigned to the DF relaying scheme to transmit the offloaded raw data, which is required to be further processed by the MERS. On the other hand, the remaining fraction is assigned to the AF relaying scheme to transmit the locally computed results without any additional procedure involved in the computations.

The proposed HR architecture offers several advantages over the existing AF and DF relaying schemes, as it can inherit the benefits of both AF and DF schemes. In particular, the AF scheme tends to be superior at low signal-to-noise ratios (SNRs) due to its low computational complexity, reduced energy consumption and delay, whilst avoiding the error-propagation routinely encountered by DF relaying at low SNRs. By contrast, the DF scheme tends to perform better at high SNRs, potentially eliminating all errors [15]. Hence, by combing the merits of AF and DF relaying schemes to enhance

---

[1]DVFS is a technique that varies the supply voltage and clock frequency of a processor based on the computation load, in order to provide the desired computation performance while reducing energy consumption. (DVFS) by considering either energy or latency minimization.

the system performance, the HR architecture is suitable for a wider range of applications. Moreover, the additional design flexibility provided by the proposed HR architecture allows more efficient resource allocation, leading in turn to reduced execution delay and energy consumption (as demonstrated later). We emphasize that in our proposed approach, there is no need to explicitly carry out relay selection as it will be automatically determined by the optimal offloading ratio.

Within this framework, we seek to minimize the weighted sum of the execution delays and the energy consumption in the RACO system by jointly optimizing the computation offloading ratio, the processor clock rate, the bandwidth allocation between the AF and the DF schemes, as well as the transmit power levels of user A and the MERS, under the practical constraints imposed by the available computing and communication resources. The formulated problem is very challenging due to the highly coupled and non-differentiable nature of the objective function and constraints. Still, by exploiting the structure of the problem and invoking the concave-convex procedure (CCCP) [21], we devise efficient joint resource allocation algorithms to solve it.

Against this background our main contributions can be summarized as follows:

1) A new HR architecture is proposed for RACO system in MEC applications, i.e., to support computation offloading as well as the transfer of locally computed results. For this architecture, we formulate a joint resource allocation problem aiming to minimize the weighted sum of the execution delays and the energy consumption, subject to a number of realistic constraints.

2) By applying a series of suitable transformations and introducing auxiliary variables, we recast this challenging optimization problem into an equivalent but more tractable form. For the resultant problem, we develop a new CCCP-based algorithm to handle the highly coupled terms and to jointly optimize the RACO system parameters.

3) By further exploiting the problem structure, an efficient low-complexity algorithm is proposed based on the smooth approximation and the inexact block coordinate descent (IBCD) method of optimization.

4) To gain additional insights into the proposed RACO system with HR architecture, we consider two special cases for the off-loading ratio, respectively leading to AF only and DF only schemes. For these cases, the optimization problem is further simplified and a pair of further algorithmic solutions are proposed.

5) Finally, we present and discuss our simulation results to shed more lights on both the convergence properties and the overall performance of our schemes and algorithms proposed for RACO in MEC applications.

This paper is structured as follows. Section II describes the proposed RACO system model with HR architecture and formulates the resource allocation problem of interest. Section III transforms the original problem into a more tractable yet equivalent form and develops a CCCP-based algorithm for its solution. In Section IV, a smooth approximation is applied and a simplified algorithm based on the IBCD method is derived.

Section V considers the special cases of the AF only and the DF only, and for each case, proposes simplified and efficient solutions. Section VI presents the simulation results. Finally, the paper is concluded in Section VII.

II. SYSTEM MODEL AND PROBLEM STATEMENT

We consider the RACO system with HR architecture illustrated in Fig. 1, where user A aims to share the results of computational tasks with user B through wireless exchange over a MERS, i.e., a relay platform equipped with MEC capabilities. Due to hardware limitations, user A only performs a fraction of its tasks locally while the remaining fraction is transferred to the MERS, where more extensive resources are available. To support the computation offloading, we propose an HR architecture employing a combination of the AF and the DF relay schemes over the entire available bandwidth.

Specifically, the AF scheme is used over a selected portion of the bandwidth to relay the computational results of user A to user B, while the DF scheme is used over the remaining bandwidth to offload the computational tasks of user A to the MERS and then to relay the results to user B. Clearly, depending on the task offloading ratio, the bandwidth allocation ratio, the available computing resources and the transmit power, user A's offloading strategy may significantly affect both the end-to-end delay and the system's overall energy consumption. In this work, our aim is to balance these two system performance metrics by appropriately allocating both the computational and communication resources.

Denoting the total available spectral bandwidth by W (in Hz), let $\nu \in [0, 1]$ denote the fraction of this bandwidth allocated to the DF scheme, so that $1 - \nu$ is the fraction allocated to the AF scheme. Let $h_A^{(1)}$ and $h_B^{(1)}$ respectively denote the AF relay channels between user A and the relay, as well as between the relay and user B. Similarly, let $h_A^{(2)}$ and $h_B^{(2)}$ respectively denote the DF relay channels between user A and the relay, as well as between the relay and user B. Moreover, assume that partial offloading is implemented on the basis of the *data partitioning-oriented tasks* of [13], [22], and that the size of the computational results are proportional to the size of the input tasks. We characterize the computational tasks at user A by the triplet (L, K, ρ), where L (in bits) denotes the size of the tasks before computation, K denotes the average number of central processing unit (CPU) cycles required for processing each bit, and $\rho > 0$ denotes the conversion ratio between the size of the tasks before computation and the size of the corresponding results after computation. Finally, let $\alpha \in [0, 1]$ denote the fraction of computational tasks offloaded by user A to the MERS, i.e.: αL bits are transferred to the MERS via the DF channel for remote processing, while $(1 - \alpha)L$ bits are processed locally, and the results then being forwarded to the MERS via the AF channel. Note that in this paper, we focus on the resource allocation of our RACO system under the half-duplex setup and do not restrict the duplex mode to be time-division duplexing (TDD) and frequency-division duplexing (FDD). The extension to a more complex full-duplex setup, is however beyond the scope of this manuscript and it is set aside for our future work.
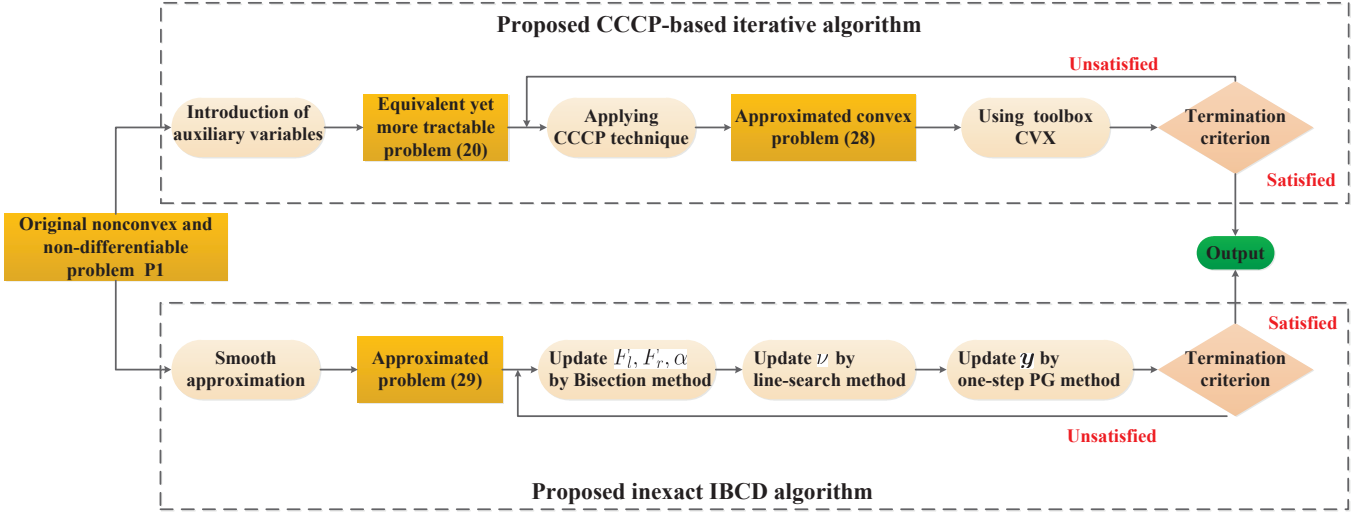
Fig. 2: Development of efficient resource allocation policies for the RACO system.

Let us first consider the AF relaying scheme. In this case, the signal received at the MERS is given by

$$y_R^{(1)} = \sqrt{P_1^A} h_A^{(1)} \overline{x}_{A1} + n_R^{(1)}, \tag{1}$$

where $\overline{x}_{A1} \sim \mathcal{CN}(0,1)$ denotes the transmit signal after local computing, $n_R^{(1)}$ denotes the complex additive white Gaussian noise (AWGN) at the MERS of power spectral density (PSD) σ_{R1}^2, and P_1^A denotes the transmit power of user A. The MERS amplifies the received signal and forwards it to user B. Therefore, the signal received at user B is given by

$$
\begin{aligned}
y_B^{(1)} &= \sqrt{P_1^R} h_B^{(1)} y_R^{(1)} + n_B^{(1)} \\
&= \sqrt{P_1^R}\sqrt{P_1^A} h_B^{(1)} h_A^{(1)} \overline{x}_{A1} + \sqrt{P_1^R} h_B^{(1)} n_R^{(1)} + n_B^{(1)}, \quad (2)
\end{aligned}
$$

where $n_B^{(1)} \sim \mathcal{CN}(0,\sigma_{B1}^2)$ and P_1^R denote the AWGN at user B and the transmit power of the MERS allocated to the AF relaying scheme, respectively. According to (2), the transmission rate and delay in the AF scheme are expressed as

$$R_{\text{AF}} \triangleq \frac{(1-\nu)W}{2} \log_2 \left(1 + \frac{P_1^A P_1^R |h_B^{(1)} h_A^{(1)}|^2}{P_1^R |h_B^{(1)}|^2 \sigma_{R1}^2 + \sigma_{B1}^2}\right), \quad (3)$$

$$t_{\text{AF}} \triangleq \frac{(1-\alpha)\rho L}{R_{\text{AF}}}. \tag{4}$$

Note that the pre-log factor $\frac{1}{2}$ in (3) is induced by the half-duplex signaling (two channel uses) and causes a loss in spectral efficiency [23], [24]. Furthermore, the energy consumption of the AF scheme is given by

$$
\begin{aligned}
E_{AF} &\triangleq (P_1^A + P_1^R \mathbb{E}[|y_R^{(1)}|^2]) t_{\text{AF}} \\
&= (P_1^A + P_1^R P_1^A |h_A^{(1)}|^2 + P_1^R \sigma_{R1}^2) t_{\text{AF}}. \quad (5)
\end{aligned}
$$

Next, let us consider the DF relaying scheme. In this case, user A first offloads its unprocessed computational tasks to the MERS, where they are decoded and processed. Similar to the AF relaying scheme, the signal received at the MERS is

$$y_R^{(2)} = \sqrt{P_2^A} h_A^{(2)} x_{A2} + n_R^{(2)}, \tag{6}$$

where $x_{A2} \sim \mathcal{CN}(0,1)$ denotes the transmit signal from user A, $n_R^{(2)}$ denotes the AWGN at the MERS of PSD σ_{R2}^2, and P_2^A denotes the transmit power of user A in the DF relay scheme. The transmission rate and delay for the first hop in the DF scheme are given by

$$R_{\text{DF1}} \triangleq \nu W \log_2 \left(1 + \frac{P_2^A |h_A^{(2)}|^2}{\sigma_{R2}^2}\right), \tag{7}$$

$$t_{\text{DF1}} \triangleq \frac{\alpha L}{R_{\text{DF1}}}. \tag{8}$$

After decoding the message from A, the MERS executes the offloaded processing tasks and then re-encodes and forwards the computational results to user B. As in [24], [25], it is assumed that the processing delay and energy consumption associated to the decoding and encoding operations at the MERS are negligible compared to those of edge computing, i.e. processing user A's computational tasks. The signal received at user B is given by

$$y_B^{(2)} = \sqrt{P_2^R} h_B^{(2)} \overline{x}_{A2} + n_B^{(2)}, \tag{9}$$

where $\overline{x}_{A2} \sim \mathcal{CN}(0,1)$ denotes the transmit signal by the MERS after edge computing, $n_B^{(2)}$ denotes the complex AWGN at the destination of PSD σ_{B1}^2, and P_2^R denotes the transmit power of MERS in the DF relaying scheme. According to (9), the rate and delay for the second hop in the DF scheme can be expressed as

$$R_{\text{DF2}} \triangleq \nu W \log_2 \left(1 + \frac{P_2^R |h_B^{(2)}|^2}{\sigma_{B2}^2}\right) \tag{10}$$

$$t_{\text{DF2}} \triangleq \frac{\alpha \rho L}{R_{\text{DF2}}}. \tag{11}$$

Furthermore, the corresponding energy consumption is

$$E_{DF} \triangleq P_2^A t_{\text{DF1}} + P_2^R t_{\text{DF2}}. \tag{12}$$

As in [13], we model the power consumption of a generic CPU as $P = \eta F^3$, where F denotes the CPU's computation speed (in cycles per second) and η is a coefficient depending

on the chip architecture; hence, the energy consumption per cycle is given by ηF^2. For local computation, the energy consumption can be minimized by optimally configuring computation speed via the DVFS technology. Hence, if the amount of data bits processed at user A is $(1-\alpha)L$, the execution time t_l will be $t_l \triangleq K_l(1-\alpha)L/F_l$, where F_l denotes the computation speed of user A, and the corresponding energy consumption E_l is given by $E_l \triangleq (1-\alpha)LK_l\eta_l F_l^2$.

Similarly, the execution time and the energy consumption of edge computation are respectively given by $t_r \triangleq K_r\alpha L/F_r$, and $E_r \triangleq \alpha LK_r\eta_r F_r^2$, where F_r denotes the computation speed of the MERS.

Considering both AF and DF relaying schemes, the total latency for executing the computational tasks of user A within the RACO framework is given by

$$t_{\text{sys}} \triangleq \max\{t_l + t_{\text{AF}}, t_{\text{DF1}} + t_r + t_{\text{DF2}}\}, \qquad (17)$$

and the system's total energy consumption is expressed as

$$E_{\text{sys}} \triangleq E_l + E_r + E_{AF} + E_{DF}. \qquad (18)$$

Our interest in this work lies in finding efficient algorithmic solutions to the following resource allocation problem, referred to as the constrained *weighted sum of execution delays and energy consumption minimization* problem[2]:

$$\mathbf{P1} : \min_{\mathbf{x}} E_{\text{sys}} + \gamma t_{sys} \qquad (19\text{a})$$

$$\text{s.t.} \quad 0 < F_l \le F_l^{\max}, \qquad (19\text{b})$$

$$0 < F_r \le F_r^{\max}, \qquad (19\text{c})$$

$$0 \le \alpha \le 1, \qquad (19\text{d})$$

$$0 \le \nu \le 1, \qquad (19\text{e})$$

$$P_i^A \ge 0, \qquad (19\text{f})$$

$$P_i^R \ge 0, \qquad (19\text{g})$$

$$\sum_{i=1}^{2} P_i^A \le P_A^{\max}, \qquad (19\text{h})$$

$$P_1^R\sigma_{R1}^2 + |h_A^{(1)}|^2 P_1^R P_1^A + P_2^R \le P_R^{\max}, \qquad (19\text{i})$$

where $\mathbf{x} \triangleq [\alpha, \nu, P_1^A, P_2^A, P_1^R, P_2^R, F_l, F_r]^T$ denotes the vector of search variables. The objective function in $\mathbf{P1}$ is a weighted sum of the system's total latency and energy consumption, where the weighting factor γ (in Joule/sec) allows a proper trade-off between these two key metrics. Constraints (19b) and (19c) are the maximum computation speed constraints imposed by user A's and the MERS's CPUs, respectively. Constraints (19f), (19g), (19h) and (19i) specify the transmission power budgets at user A and the MERS.

Remark 1: In a practical 3GPP-LTE network [26], the transmit power is determined based on each resource block and remains constant within each resource block. Since MERS is deployed in the proximity of user A and equipped with ample computational and storage resources [7], the delays encountered both by AF and DF relaying at the MERS are low. In other words, both the AF and the DF operations at

[2]For constraints involving user index i, it is implicitly assumed that the constraint must apply $\forall i \in \{1, 2\}$.

the MERS can be carried out within the same resource block, which makes (19i) reasonable.

Due to the nonconvex and non-differentiable objective function in (19a), along with the nonconvex coupling constraint (19h), the solution of problem $\mathbf{P1}$ remains challenging. Based on our problem formulation $\mathbf{P1}$, we will develop optimization approaches for devising efficient resource allocation policies for the RACO system. The main flow of ideas in developing these policies is illustrated in Fig. 2 as displayed at the top of this page, while their precise details will be explained in the following sections.

III. PROPOSED CCCP BASED ALGORITHM

In this section, we first transform our problem $\mathbf{P1}$ into an equivalent yet more tractable form by introducing auxiliary variables, and subsequently develop an efficient CCCP based algorithm to solve the transformed problem. To this end, a locally tight upper bound is derived to obtain a convex approximation to the objective function, while linearization is applied to approximate the nonconvex constraints.

A. Problem Transformation

We first transform problem $\mathbf{P1}$ into an equivalent but more tractable form. Specifically, by introducing a number of auxiliary variables represented by the vector $\boldsymbol{\phi} \triangleq [t_s, t_A, t_{D1}, t_{D2}, t_l, t_r, R_A, R_{D1}, R_{D2}, \Lambda_A, \Lambda_{D2}, \Lambda_{D2}, \varphi_1, \varphi_2, \varphi_3, s_1, s_2]^T$, problem $\mathbf{P1}$ can be formulated as the following equivalent problem:

$$\mathbf{P2} : \min_{\mathbf{x},\boldsymbol{\phi}} E_{\text{sys}} + \gamma t_s \qquad (20\text{a})$$

$$\text{s.t.} \quad (1-\alpha)\rho L \le t_A R_A, \ 2R_A \le (1-\nu)W\Lambda_A, \quad (20\text{b})$$

$$\Lambda_A \le \log_2(1+1/\varphi_1), \ \alpha L \le t_{D1}R_{D1}, \qquad (20\text{c})$$

$$R_{D1} \le \nu W\Lambda_{D1}, \ \Lambda_{D1} \le \log_2(1+1/\varphi_2), \qquad (20\text{d})$$

$$\alpha\rho L \le R_{D2}t_{D2}, \ R_{D2} \le \nu W\Lambda_{D2}, \qquad (20\text{e})$$

$$\Lambda_{DF2} \le \log_2(1+1/\varphi_3), \sigma_{R2}^2 - \varphi_2 P_2^A|h_A^{(2)}|^2 \le 0, \qquad (20\text{f})$$

$$|h_B^{(1)}|^2\sigma_{R1}^2 P_1^R + \sigma_{B1}^2 - |h_B^{(1)}h_A^{(1)}|^2\varphi_1 s_2 \le 0, \quad (20\text{g})$$

$$\sigma_{B2}^2 - \varphi_3 P_2^R|h_B^{(2)}|^2 \le 0, s_2 \le P_1^A P_1^R \le s_1, \quad (20\text{h})$$

$$K_l(1-\alpha)L \le t_l F_l, \ K_r\alpha L \le t_r F_r, \qquad (20\text{i})$$

$$t_l + t_A \le t_s, \ t_{D1} + t_r + t_{D2} \le t_s, \qquad (20\text{j})$$

$$(19\text{b}) - (19\text{h}), \qquad (20\text{k})$$

where

$$E_{\text{sys}} \triangleq (1-\alpha)LK_l\eta_l F_l^2 + \alpha LK_r\eta_r F_r^2 t_A + P_2^A t_{D1} + P_2^R t_{D2} + (P_1^A + P_1^R\sigma_{R1}^2 + s_1|h_A^{(2)}|^2) \qquad (21)$$

denotes the system energy consumption. Note that problem $\mathbf{P1}$ and $\mathbf{P2}$ share the same global optimal solution for $\mathbf{x}$ under the given constraints. The detailed derivation of the equivalence between problems $\mathbf{P1}$ and $\mathbf{P2}$ is presented in Appendix A.

B. Algorithm Proposed for Solving Problem **P2**

In this part, we propose an efficient CCCP-based algorithm for solving problem **P2**. In order to approximate this problem as a convex one, we first find a locally tight upper bound of the objective and then linearize the nonconvex constraints with the aid of the CCCP concept, so that the nonconvex problem **P2** can be approximated as a convex one.

1) Upper Bound for the Objective Function: Our approach for bounding the objective function in problem P2 relies on the following lemma.

Lemma 1: [27] Supposes that g has a separable structure as the product of two convex and non-negative real-valued functions h_1 and h_2, that is, $g(\boldsymbol{x}_1, \boldsymbol{x}_2) = h_1(\boldsymbol{x}_1)h_2(\boldsymbol{x}_2)$. For any $(\boldsymbol{y_1}, \boldsymbol{y_2})$ in the domain of g, a convex approximation of $g(\boldsymbol{x}_1, \boldsymbol{x}_2)$ in the neighborhood of $(\boldsymbol{y_1}, \boldsymbol{y_2})$, which satisfies mild conditions required by the CCCP algorithm, is defined in (22) as displayed at the bottom of this page.

Note that the continuity and smoothness conditions of the CCCP requires the strongly convex approximation of the objective function to have the same first derivative as the objective function, while the convex approximations of the constraints are required to be tight at the point of interest and to bound the original constraints. Based on *Lemma 3.1*, we can obtain a locally tight upper bound for the objective function of problem **P2** in the current iteration as follows,

$$f_3(\mathbf{x}, \boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}}) \triangleq f_1(\mathbf{x}, \boldsymbol{\phi}) - \hat{f}_2(\mathbf{x}, \boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}}). \tag{23}$$

where $\tilde{\boldsymbol{x}} \triangleq [\tilde{\alpha}, \nu, \tilde{P}_1^A, \tilde{P}_2^A, \tilde{P}_1^R, \tilde{P}_2^R, \tilde{F}_l, \tilde{F}_r]^T$, and $\tilde{\boldsymbol{\phi}} \triangleq [\tilde{t}_s, \tilde{t}_A, \tilde{t}_{D1}, \tilde{t}_{D2}, \tilde{t}_l, \tilde{t}_r, \tilde{R}_A, \tilde{R}_{D1}, \tilde{R}_{D2}, \tilde{\varphi}_1, \tilde{\varphi}_2, \tilde{\varphi}_3, \tilde{s}_1, \tilde{s}_2]^T$ are the current points generated by the last iteration; $f_1(\mathbf{x}, \boldsymbol{\phi})$ and $\hat{f}_2(\mathbf{x}, \boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}})$ are respectively defined in (24)-(25) as displayed at the bottom of this page.

Please refer to Appendix B for the constructive derivation.

2) Linearizing the Nonconvex Constraints: Note that all the nonconvex constraints in problem **P2** have a similar structure and can be equivalently converted to convex constraints. Here, we focus on the conversion of the first constraint in (20b) as an example. By applying the equality $xy = \frac{1}{2}[(x+y)^2 - x^2 - y^2]$, we can rewrite this constraint into the DC program:

$$\underbrace{2(1-\alpha)\rho L + t_{\text{AF}}^2 + R_{\text{AF}}^2}_{\text{convex function}} - \underbrace{(R_{\text{AF}} + t_{\text{AF}})^2}_{\text{convex function}} \leq 0, \tag{26}$$

Algorithm 1: The CCCP-based iterative algorithm

0. **Initialization**: Define the tolerance of accuracy δ and the maximum number of iterations $N_{\max}$. Initialize the algorithm with a feasible point $\mathbf{x}^0, \boldsymbol{\phi}^0$. Set the iteration number $i = 0$.
1. **repeat**
2.   Solve the convex optimization problem (28) with the affine approximation, and assign the solution to $\mathbf{x}^{i+1}, \boldsymbol{\phi}^{i+1}$.
3.   Update the iteration number: $i \leftarrow i + 1$
4. **until** $|f_3(\mathbf{x}^i, \boldsymbol{\phi}^i) - f_3(\mathbf{x}^{i-1}, \boldsymbol{\phi}^{i-1})| \leq \delta$ or reaching the maximum iteration number.

By linearizing the subtracted convex terms in (26) by applying the first-order Taylor expansion around the current point $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}})$, we obtain

$$2(1-\alpha)\rho L + t_{\text{A}}^2 + R_{\text{A}}^2 - (\tilde{R}_{\text{A}} + \tilde{t}_{\text{A}})^2$$
$$- 2(\tilde{R}_{\text{A}} + \tilde{t}_{\text{A}})(R_{\text{A}} + t_{\text{A}} - \tilde{R}_{\text{A}} - \tilde{t}_{\text{A}}) \leq 0. \tag{27}$$

The other constraints in problem **P2** can be converted by a similar process, but the details are omitted due to space limitation. Finally, based on the CCCP concept, problem **P2** can be reformulated as an iterative sequence of convex optimization problems defined in (28) as displayed at the bottom of the next page. Problem (28) can be efficiently solved by the convex programming toolbox CVX [28]. The implementation of the proposed CCCP-based iterative procedure is summarized in Algorithm 1.

C. Convergence Analysis and Computational Complexity

Repeated application of the CCCP iteration will eventually lead to a stationary solution of problem **P2** [29]. We can show that the limit point of the iterations generated by the proposed CCCP algorithm also satisfies the KKT conditions of the DC program (28), which guarantees convergence to a locally optimal solution of problem **P1**. The proof is similar to that of *Lemma 2* and *Theorem 1* in [30], hence we omit the details.

The overall computational complexity of Algorithm 1 is dominated by the interior point method implemented by CVX toolbox, which is significantly affected by the number of second-order cone (SOC) constraints of problem (28) and the corresponding dimensions. To access the complexity, we can transform the constraints of problem (28) into the form of SOC

$$\hat{g}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2) \triangleq \frac{1}{2}(h_1(\boldsymbol{x}_1) + h_2(\boldsymbol{x}_1))^2 - \frac{1}{2}(h_1(\boldsymbol{y}_1) + h_2(\boldsymbol{y}_2))^2 - h_1'(\boldsymbol{y}_1)h_1(\boldsymbol{y}_1)(\boldsymbol{x}_1 - \boldsymbol{y}_1) - h_2'(\boldsymbol{y}_2)h_2(\boldsymbol{y}_2)(\boldsymbol{x}_2 - \boldsymbol{y}_2). \tag{22}$$

$$f_1(\mathbf{x}, \boldsymbol{\phi}) \triangleq \frac{1}{2}[2LK_l\eta_l F_l^2 + LK_r\eta_r(\alpha + F_r^2)^2 + LK_l\eta_l(\alpha^2 + F_l^4) + (P_1^A + t_{\text{A}})^2 + \sigma_{R1}^2(P_1^R + t_{\text{A}})^2$$
$$+ |h_A^{(1)}|^2(s_1 + t_{\text{A}})^2 + (P_2^A + t_{\text{D1}})^2 + (P_2^R + t_{\text{D2}})^2] + \gamma t_s, \tag{24}$$

$$\hat{f}_2(\mathbf{x}, \boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}}) = \frac{1}{2}[LK_r\eta_r(\tilde{\alpha}^2 + \tilde{F}_r^4) + LK_l\eta_l(\tilde{\alpha} + \tilde{F}_l^2)^2 + LK_r\eta_r\tilde{\alpha}(\alpha - \tilde{\alpha}) + 2LK_r\eta_r\tilde{F}_r^3(F_r - \tilde{F}_r) + LK_l\eta_l(\tilde{\alpha} + \tilde{F}_l^2)(\alpha - \tilde{\alpha})$$
$$+ 2LK_l\eta_l\tilde{F}_l(\tilde{\alpha} + \tilde{F}_l^2)(F_l - \tilde{F}_l)] + (\tilde{P}_1^A)^2 + \tilde{t}_{\text{A}}^2 + \sigma_{R1}^2((\tilde{P}_1^R)^2 + \tilde{t}_{\text{A}}^2) + |h_A^{(1)}|^2(\tilde{s}_1^2 + \tilde{t}_{\text{AF}}^2) + (\tilde{P}_2^A)^2 + \tilde{t}_{\text{D1}}^2$$
$$+ (\tilde{P}_2^R)^2 + \tilde{t}_{\text{D2}}^2] + [\tilde{P}_1^A(P_1^A - \tilde{P}_1^A) + \tilde{t}_{\text{A}}(t_{\text{A}} - \tilde{t}_{\text{A}}) + \sigma_{R1}^2(\tilde{P}_1^R(P_1^R - \tilde{P}_1^R) + \tilde{t}_{\text{A}}(t_{\text{A}} - \tilde{t}_{\text{A}})) + \tilde{P}_2^A(P_2^A - \tilde{P}_2^A)$$
$$+ |h_A^{(1)}|^2(\tilde{s}_1(s_1 - \tilde{s}_1) + \tilde{t}_{\text{A}}(t_{\text{A}} - \tilde{t}_{\text{A}})) + \tilde{t}_{\text{D1}}(t_{\text{D1}} - \tilde{t}_{\text{D1}}) + \tilde{P}_2^R(P_2^R - \tilde{P}_2^R) + \tilde{t}_{\text{D2}}(t_{\text{D2}} - \tilde{t}_{\text{D2}})]. \tag{25}$$

(details are omitted due to space limitation). In a nutshell, problem (28) contains 7 SOC constraints of dimension 3 while the number of optimized variables is 21. Hence, the number of required floating point operations (FPOs) at each iteration is on the order of 10^5. The complexity of the proposed CCCP-based algorithm is given by the number of required FPOs $n_1 = M_1 I_1$, where I_1 denotes the number of required iterations and M_1 denotes the number of FPOs at each iteration.

IV. PROPOSED LOW COMPLEXITY ALGORITHM FOR PROBLEM **P1**

The proposed CCCP-based algorithm can be applied to address problem **P1** but incurs a very high computational complexity due to the need to solve the sequence of convex optimization problems (28). In this section, by further exploiting the problem structure, we propose an alternative iterative algorithm with much reduced complexity. Specifically, we first approximate the objective function of P1 as a smooth function and then propose an inexact BCD algorithm (a variant of the BCD algorithm [41]) to solve the resulting problem.

A. Smooth Approximation of Objective Function

Observing that the constraints in **P1** are separable w.r.t. the five blocks of variables, i.e., F_l, F_r, α, ν, and $\mathbf{y} \triangleq \{P_1^A, P_2^A, P_1^R, P_2^R\}$, we may apply the IBCD algorithm to problem **P1**. This requires the objective function to be differentiable, which is not the case here due to (17). To address the non-differentiability issue, we first approximate the objective function of **P1** by a smooth function using the log-smooth

method. Specifically, using the log-sum-exp inequality [34, pp. 72], we have

$$\max(x,y) \le \frac{1}{\beta} \log(\exp(\beta x) + \exp(\beta y)) \le \max(x,y) + \frac{1}{\beta} \log 2. \tag{29}$$

Utilizing (29), we can approximate t_{sys} as

$$\hat{t}_{\text{sys}} \approx \frac{1}{\beta} \log(\exp(\beta(t_l + t_{\text{AF}})) + \exp(\beta(t_{\text{DF1}} + t_r + t_{\text{DF2}})))$$

with a large β. Problem **P1** is smoothly approximated as

$$\min_{\mathbf{x}} \ f_\beta(\mathbf{x}) \quad \text{s.t.} \quad (19\text{b}) - (19\text{h}), \tag{30}$$

where $f_\beta(\mathbf{x}) \triangleq E_{\text{sys}} + \gamma \hat{t}_{sys}$ denotes the approximated objective function of problem **P1** with the differentiable property.

B. Inexact Block Coordinate Algorithm for Smoothed Problem

We can now use the IBCD method to solve the smoothened problem (30). In this method, we sequentially update each block of variables, while fixing the other blocks to their previous values. For problem (30), this amounts to the following steps:

Step 1: Updating F_l while fixing $\{F_r, \alpha, \nu, \mathbf{y}\}$. Let us now consider the subproblem w.r.t. F_l, which is given by

$$\min_{0 < F_l \le F_l^{\max}} E_l + \gamma \hat{t}_{sys} \tag{31}$$

It can be verified that the above subproblem is convex, and thus can be easily solved using the bisection method [34].

Step 2: Updating F_r while fixing $\{F_l, \alpha, \nu, \mathbf{y}\}$. Similar to the subproblem w.r.t. F_l, the subproblem w.r.t F_r is also convex and thus can be solved using the bisection method.

$$\min_{\mathbf{x}, \phi} f_3(\mathbf{x}, \phi; \tilde{\mathbf{x}}, \tilde{\phi}) \tag{28a}$$

$$\text{s.t.} \ \Lambda_A - \log_2(1 + 1/\tilde{\varphi}_1) + (\varphi_1 - \tilde{\varphi}_1)/[\ln 2((\tilde{\varphi}_1)^2 + \tilde{\varphi}_1)] \le 0, \tag{28b}$$

$$2R_A + W(\nu + \Lambda_A)^2/2 - W\Lambda_A - W((\tilde{\nu})^2 + (\tilde{\Lambda}_A)^2)/2 - W\tilde{\nu}(\nu - \tilde{\nu}) - W\tilde{\Lambda}_A(\Lambda_A - \Lambda_A) \le 0, \tag{28c}$$

$$2\alpha L + t_{\text{D1}}^2 + R_{\text{D1}}^2 - (\tilde{R}_{\text{D1}} + \tilde{t}_{\text{D1}})^2 - 2(\tilde{R}_{\text{D1}} + \tilde{t}_{\text{D1}})(R_{\text{D1}} + t_{\text{D1}} - \tilde{R}_{\text{D1}} - \tilde{t}_{\text{D1}}) \le 0, \tag{28d}$$

$$R_{\text{D1}} + W(\nu^2 + \Lambda_{\text{D1}}^2)/2 - W(\tilde{\nu} + \tilde{\Lambda}_{\text{D1}})^2/2 - W(\tilde{\nu} + \tilde{\Lambda}_{\text{D1}})(\nu + \Lambda_{\text{D1}} - \tilde{\nu} - \tilde{\Lambda}_{\text{D1}}) \le 0, \tag{28e}$$

$$\Lambda_{\text{D1}} - \log_2(1 + 1/\tilde{\varphi}_2) + (\varphi_2 - \tilde{\varphi}_2)/[\ln 2((\tilde{\varphi}_2)^2 + \tilde{\varphi}_2)] \le 0, \tag{28f}$$

$$2\sigma_{R2}^2 + |h_A^{(2)}|^2[\varphi_2^2 + P_2^{A^2} - (\tilde{\varphi}_2 + \tilde{P}_2^A)^2 - 2(\tilde{\varphi}_2 + \tilde{P}_2^A)(\varphi_2 + P_2^A - \tilde{\varphi}_2 - \tilde{P}_2^A)] \le 0, \tag{28g}$$

$$2\alpha\rho L + t_{\text{D2}}^2 + R_{\text{D2}}^2 - (\tilde{R}_{\text{D2}} + \tilde{t}_{\text{D2}})^2 - 2(\tilde{R}_{\text{D2}} + \tilde{t}_{\text{D2}})(R_{\text{D2}} + t_{\text{D2}} - \tilde{R}_{\text{D2}} - \tilde{t}_{\text{D2}}) \le 0, \tag{28h}$$

$$R_{\text{D1}} + W(\nu^2 + \Lambda_{\text{D2}}^2)/2 - W(\tilde{\nu} + \tilde{\Lambda}_{\text{D2}})^2/2 - W(\tilde{\nu} + \tilde{\Lambda}_{\text{D2}})(\nu + \Lambda_{\text{D2}} - \tilde{\nu} - \tilde{\Lambda}_{\text{D2}}) \le 0, \tag{28i}$$

$$\Lambda_{\text{D2}} - \log_2(1 + 1/\tilde{\varphi}_3) + (\varphi_3 - \tilde{\varphi}_3)/[\ln 2((\tilde{\varphi}_3)^2 + \tilde{\varphi}_3)] \le 0, \tag{28j}$$

$$2\sigma_{B2}^2 + |h_B^{(2)}|^2[\varphi_3^2 + P_2^{R^2} - (\tilde{\varphi}_3 + \tilde{P}_2^R)^2 - 2(\tilde{\varphi}_3 + \tilde{P}_2^R)(\varphi_3 + P_2^R - \tilde{\varphi}_3 - \tilde{P}_2^R)] \le 0, \tag{28k}$$

$$(P_1^R + P_1^A)^2 - (\tilde{P}_1^R)^2 - (\tilde{P}_1^A)^2 - 2\tilde{P}_1^R(P_1^R - \tilde{P}_1^R) - 2\tilde{P}_1^A(P_1^A - \tilde{P}_1^A) - 2s_1 \le 0, \tag{28l}$$

$$2s_2 + P_1^{R^2} + P_1^{A^2} - (\tilde{P}_1^A + \tilde{P}_1^R)^2 - 2(\tilde{P}_1^A + \tilde{P}_1^R)(P_1^A + P_1^R - \tilde{P}_1^A - \tilde{P}_1^R) \le 0, \tag{28m}$$

$$2K_l(1 - \alpha)L + F_l^2 + t_l^2 - (\tilde{t}_l + \tilde{F}_l)^2 - 2(\tilde{t}_l + \tilde{F}_l)(t_l + F_l - \tilde{t}_l - \tilde{F}_l) \le 0, \tag{28n}$$

$$2K_r\alpha L + F_r^2 + t_r^2 - (\tilde{t}_r + \tilde{F}_r)^2 - 2(\tilde{t}_r + \tilde{F}_r)(t_r + F_r - \tilde{t}_r - \tilde{F}_r) \le 0, \tag{28o}$$

$$(19\text{b}) - (19\text{h}), (20\text{j}), (27). \tag{28p}$$

Step 3: Updating α while fixing $\{F_l, F_r, \nu, \mathbf{y}\}$. The subproblem w.r.t. α is also convex and thus can be solved using bisection.

Step 4: Updating ν while fixing $\{F_l, F_r, \alpha, \mathbf{y}\}$. Since the subproblem w.r.t. ν is the minimization of a scalar function, it can be efficiently solved by the line-search method with the well-known Armijo backtracking step size [34].

Step 5: Updating $\mathbf{y}$ while fixing $\{F_l, F_r, \nu, \alpha\}$. Let us consider the subproblem w.r.t. $\mathbf{y}$, which is given by

$$\min_{\mathbf{y}} f_\beta(\mathbf{y}) \quad \text{s.t.} \quad (19e) - (19h). \tag{32}$$

Obviously, (19h) is a nonconvex constraint, which complicates the solution of (32). To efficiently update $\mathbf{y}$ while reducing the objective value, we apply the concept of linearization to tackle the nonconvexity of (19h). First, we express the latter as a DC program:

$$P_2^R + P_1^R \sigma_{R1}^2 + \frac{1}{2}|h_A^{(1)}|^2[(P_1^R + P_1^A)^2 - (P_1^R)^2 - (P_1^A)^2] \le P_R^{\max}. \tag{33}$$

By linearizing the nonconvex term $-(P_1^R)^2 - (P_1^A)^2$ at the current point $\tilde{\mathbf{y}} \triangleq [\tilde{P}_1^A, \tilde{P}_2^A, \tilde{P}_1^R, \tilde{P}_2^R]^T$, we approximate (33) as a convex constraint

$$U(\mathbf{y}; \tilde{\mathbf{y}}) \triangleq P_2^R + P_1^R \sigma_{R1}^2 + \frac{1}{2}|h_A^{(1)}|^2[(P_1^R + P_1^A)^2 - (\tilde{P}_1^R)^2 - (\tilde{P}_1^A)^2 - 2\tilde{P}_1^R P_1^R - 2\tilde{P}_1^A P_1^A] - P_R^{\max} \le 0. \tag{34}$$

As a result, we can approximate problem (32) as

$$\min_{\mathbf{y}} f_\beta(\mathbf{y}) \quad \text{s.t.} \quad (19e) - (19g), (34), \tag{35}$$

where the constraints are now all convex. Hence, we can apply the *one-step* projected gradient (PG) method [34] to problem (35). Specifically, we update $\mathbf{y}$ according to

$$\overline{\mathbf{y}} = P_\Omega[\tilde{\mathbf{y}} - \nabla f_\beta(\tilde{\mathbf{y}})], \tag{36}$$

$$\mathbf{y} = \tilde{\mathbf{y}} + \mu_1(\overline{\mathbf{y}} - \tilde{\mathbf{y}}), \tag{37}$$

where $\mu_1 \in [0, 1]$ can be determined by the Armijo rule, $\nabla f_\beta(\mathbf{y})$ denotes the gradient of function $f_\beta(\mathbf{y})$, Ω denotes the constraint set of problem (35), and $P_\Omega[\cdot]$ denotes the projection of the point $(\overline{\mathbf{y}} - \tilde{\mathbf{y}})$ onto Ω, i.e., the optimal solution to the following equivalent problem,

$$\min_{\overline{\mathbf{y}}} \| \overline{\mathbf{y}} - (\tilde{\mathbf{y}} - \nabla f_\beta(\tilde{\mathbf{y}})) \|^2 \tag{38a}$$

$$\text{s.t.} \quad (19e) - (19g), (34). \tag{38b}$$

Next we show how problem (38) can be globally solved using an efficient bisection method. We note that problem (38) is convex and can be solved by considering its dual problem [34]. In this regard, we define the partial Lagrangian associated with problem (38) as

$$\mathcal{L}(\overline{\mathbf{y}}, \lambda) = \| \overline{\mathbf{y}} - (\tilde{\mathbf{y}} - \nabla f_\beta(\tilde{\mathbf{y}})) \|^2 + \lambda U(\overline{\mathbf{y}}; \tilde{\mathbf{y}}), \tag{39}$$

where λ is a Lagrange multiplier. Then, the dual problem (38) can be expressed as

$$\max_{\lambda \ge 0} h(\lambda) \tag{40}$$

Algorithm 2: Proposed inexact BCD algorithm for problem (30)

---
0. Define the tolerance of accuracy ζ and the maximum number of iterations $N_{\max}$. Initialize the algorithm with a feasible point $\mathbf{x}^0 = [F_l^0, F_r^0, \alpha^0, \nu^0, \mathbf{y}^0]^T$. Set the iteration index $i=0$
1. **repeat**
2.   Perform bisection method to obtain F_l^{i+1}, F_r^{i+1}, and α^{i+1}, respectively.
3.   Perform line-search method to obtain ν^{i+1} with Armijo backtracking step size.
5.   Perform one-step PG method to obtain $\mathbf{y}^{i+1}$, using Algorithm 3 in Appendix C.
6.   Update the iteration number $i \leftarrow i + 1$
7. **until** $|f_\beta(\mathbf{x}^i) - f_\beta(\mathbf{x}^{i-1})| \le \zeta$ or reaching the maximum number of iterations.

---

where $h(\lambda)$ is the dual function given by

$$\min_{\overline{\mathbf{y}}} \mathcal{L}(\overline{\mathbf{y}}, \lambda) \quad \text{s.t.} \quad (19e) - (19g). \tag{41}$$

Note that problem (41) can be decomposed into two independent linearly constrained convex quadratic optimization subproblems w.r.t. $\{P_1^R, P_2^R\}$ and $\{P_1^A, P_2^A\}$, respectively, both of which can be globally solved in closed-form. The detailed derivation can be found in Appendix C.

Based on the above derivation, we summarize the proposed low complexity IBCD algorithm as Algorithm 2, where the five blocks of variables are sequentially updated. We note that the implementation of the one-step PG method in step 5 involves the use of Algorithm 3 presented in Appendix C.

C. Convergence Analysis and Computational Complexity

We can show that every limit point, denoted as $\mathbf{x}^*$, generated by Algorithm 2 is a stationary point of the smoothened problem, i.e. minimizing $f_\beta(\mathbf{x})$ subject to (19b)-(19h). The proof is similar to that of *Lemma 1* in [31], and it is hence omitted.

Meanwhile, the computational complexity of Algorithm 3, which dominates the one-step PG method, can be assessed by the number of FPOs $n_2 = I_2 I_3 M_2$, where I_2 denotes the number of iterations required by the main IBCD loops in Algorithm 2, I_3 denotes the number of iterations required by the Algorithm 3, and M_2 is the number of FPOs required at each iteration of Algorithm 3. Obviously, the value of M_2 is far less than M_1 in the CCCP based algorithm. Furthermore, it has been shown in [32] that the convergence rate of the PG method is $\mathcal{O}(1/I_2)$.

V. SPECIAL CASES

In this section, we investigate problem **P1** by considering the special cases $\{\alpha = 1, \nu = 1\}$ and $\{\alpha = 0, \nu = 0\}$, corresponding to the DF and AF only relay schemes, respectively. For each one of these special cases, we propose more efficient algorithmic solutions.

A. DF Relay Scheme (case $\alpha = 1$, $\nu = 1$)

Here, we focus on the scenario where user A has very limited computational resource and therefore offloads all its computational tasks to the MERS. The MERS then decodes the computing tasks, executes them using its computational

resources, re-encodes the computation results (using possibly a different codebook), and finally transmits the results to user B. Hence, user A shares computational results with user B, employing only the DF relaying scheme. Substituting $\alpha = 1$ and $\nu = 1$ into problem **P1**, we obtain

$$\min_{F_r, P_2^A, P_2^R} E_D + \gamma t_D \tag{42a}$$

$$\text{s.t.} \quad 0 < F_r \leq F_r^{\max}, \tag{42b}$$

$$0 < P_2^A \leq P_A^{\max}, \tag{42c}$$

$$0 < P_2^R \leq P_R^{\max}, \tag{42d}$$

where $E_D \triangleq LK_r\eta_r F_r^2 + LP_2^A/R_{\text{DF1}} + \rho LP_2^R/R_{\text{DF2}}$, and $t_D \triangleq K_r L/F_r + L/R_{\text{DF1}} + \rho L/R_{\text{DF2}}$. By observing that the objective function and the constraints are separable w.r.t. the three blocks of variables, F_r, P_2^A, and P_2^R, problem (42) can be decomposed into three independent problems, whose individual solutions are developed below.

1) The subproblem w.r.t. F_r: The variable F_r is updated by solving the following linearly constrained convex problem:

$$\min_{0 < F_r \leq F_r^{\max}} LK_r\eta_r F_r^2 + \gamma K_r L/F_r \tag{43}$$

Applying the first-order optimality condition yields a closed-form solution as follows,

$$F_r = \min(\max(0, (\gamma/2\eta_r)^{\frac{3}{2}}), F_r^{\max}). \tag{44}$$

In this case, we note that the optimal computation speed of the MERS, F_r, depends on the weighting factor γ and the CPU power coefficient η_r, but is independent of the size L of the computational task.

2) The subproblem w.r.t. P_2^A: We update variable P_2^A by solving the following optimization problem:

$$\min_{0 < P_2^A \leq P_A^{\max}} (P_2^A + \gamma)L/\left(W \log_2\left(1 + P_2^A|h_A^{(2)}|^2/\sigma_{R2}^2\right)\right) \tag{45}$$

It can be easily verified that this problem is non-convex, so that its direct solution remains difficult. However, by introducing auxiliary variable $u = 1/R_{\text{DF1}}$, with $R_{\text{DF1}} \equiv R_{\text{DF1}}(P_2^A)$ given by (7), problem (45) can be transformed into the following convex optimization problem,

$$\min_{u \geq G_2} G_1 u(2^{\frac{1}{Wu}} - 1) + \gamma Lu \tag{46}$$

where $G_1 \triangleq \frac{L\sigma_{R2}^2}{|h_A^{(2)}|^2}$ and $G_2 \triangleq R_{\text{DF1}}(P_A^{\max})$. Similar to the proof of *Lemma 4* in [35], we can show that the second-order derivative of the objective function of problem (46) is always greater than or equal to zero. Due to the analytic and convex nature of its objective function, problem (46) can be efficiently solved by using bisection method [34]. Given the optimal solution $u^\star$ in (46), the optimal P_2^A of (45) can be expressed as

$$P_2^A = \sigma_{R2}^2(2^{\frac{1}{Wu^\star}} - 1)/|h_A^{(2)}|^2. \tag{47}$$

3) The subproblem w.r.t. P_2^R: The variable P_2^R is updated by solving the following optimization problem:

$$\min_{0 < P_2^R \leq P_R^{\max}} \rho(P + \gamma)L/\left(W \log_2\left(1 + P_2^R|h_B^{(2)}|^2/\sigma_{B2}^2\right)\right) \tag{48}$$

It is seen that problems (48) and (45) have a similar structure and hence, by introducing auxiliary variables $v = 1/R_{\text{DF2}}$ and $R_{\text{DF2}} \equiv R_{\text{DF2}}(P_2^R)$ given in (10), (48) can be transformed into a convex optimization problem as follows,

$$\min_{v \geq G_4} G_3 v(2^{\frac{1}{Wv}} - 1) + \gamma \rho L v \tag{49}$$

where $G_3 \triangleq \rho L\sigma_{B2}^2/|h_B^{(2)}|^2$ and $G_4 \triangleq R_{\text{DF2}}(P_R^{\max})$. Problem (49) can be globally solved using an efficient bisection method. Given its optimal solution $v^\star$, the optimum solution of problem (48) is obtained as

$$P_2^R = \sigma_{B2}^2(2^{\frac{1}{Wv^\star}} - 1)/|h_B^{(2)}|^2. \tag{50}$$

B. AF Relay Scheme (case $\alpha = 0, \nu = 0$):

Here, we focus on the scenario where the MERS does not provide computing resources and all the computational tasks of user A are performed locally. The MERS then only amplifies the signal received from user A (i.e., the results of the local computations) and forwards it to user B. Hence, only the AF relaying scheme is employed in transfer of computational results from A to B. Substituting $\alpha = 0$ and $\nu = 0$ into problem **P1**, we obtain

$$\min_{F_l, P_1^A, P_1^R} E_A + \gamma t_A \tag{51a}$$

$$\text{s.t.} \quad 0 < F_l \leq F_l^{\max}, \tag{51b}$$

$$0 < P_1^A \leq P_A^{\max}, \tag{51c}$$

$$P_1^R\sigma_{R1}^2 + |h_A^{(1)}|^2 P_1^R P_1^A + P_2^R \leq P_R^{\max}, \tag{51d}$$

where $E_A \triangleq LK_l\eta_l F_l^2 + (P_1^A + P_1^R P_1^A|h_A^{(1)}|^2 + P_1^R\sigma_{R1}^2)\rho L/R_{\text{AF}}$, and $t_A \triangleq K_l L/F_l + \rho L/R_{\text{AF}}$. Observing that the constraints are separable w.r.t. the variables, i.e, F_l and $\mathbf{z} \triangleq [P_1^A, P_1^R]^T$, problem (51) can be decomposed into two independent subproblems, whose respective solutions are derived below.

1) The subproblem w.r.t. F_l: The variable F_l is updated by solving a linearly constrained convex optimization problem as follows,

$$\min_{0 < F_l \leq F_l^{\max}} LK_l\eta_l F_l^2 + \gamma K_l L/F_l \tag{52}$$

By applying the first-order optimality condition, the following closed-form solution is obtained

$$F_l = \min(\max(0, (\gamma/2\eta_l)^{\frac{3}{2}}), F_l^{\max}). \tag{53}$$

2) The subproblem w.r.t. $\mathbf{z}$: Let us consider the subproblem w.r.t. $\mathbf{z}$, which is given by

$$\min_{\mathbf{z}} f(\mathbf{z})$$

$$\text{s.t.} \quad 0 < P_1^A \leq P_A^{\max}, \tag{54a}$$

$$P_1^R\sigma_{R1}^2 + |h_A^{(1)}|^2 P_1^R P_1^A + P_2^R \leq P_R^{\max}, \tag{54b}$$

where

$$f(\mathbf{z}) = \frac{2\rho L(P_1^A + P_1^R P_1^A |h_A^{(1)}|^2 + P_1^R \sigma_{R1}^2)}{W \log_2(1 + \frac{P_1^A P_1^R |h_B^{(1)} h_A^{(1)}|^2}{P_1^R |h_B^{(1)}|^2 \sigma_{R1}^2 + \sigma_{B1}^2})}$$
$$+ \frac{2\gamma\rho L}{W \log_2(1 + \frac{P_1^A P_1^R |h_B^{(1)} h_A^{(1)}|^2}{P_1^R |h_B^{(1)}|^2 \sigma_{R1}^2 + \sigma_{B1}^2})}.$$

It can be observed that problems (54) and (35) have a similar structure. Hence, following the same approach as used for updating the variable block $\mathbf{y}$ in Section IV, we first approximate (54b) as the convex constraint (34) and apply the *one-step* PG method to (54). We update $\mathbf{z}$ according to

$$\bar{\mathbf{z}} = P_\Omega[\tilde{\mathbf{z}} - \nabla f(\tilde{\mathbf{z}})], \qquad (55)$$
$$\mathbf{z} = \tilde{\mathbf{z}} + \mu_2(\bar{\mathbf{z}} - \tilde{\mathbf{z}}), \qquad (56)$$

where $\mu_2 \in [0,1]$ can be determined by Armijo rule, $\nabla f(\tilde{\mathbf{z}})$ denotes the gradient of f, Ω denotes the constraint set of problem (54), and $P_\Omega[\cdot]$ denotes the projection of the point $(\bar{\mathbf{z}} - \tilde{\mathbf{z}})$ onto Ω, namely the optimal solution to the following problem

$$\min_{\mathbf{z}} \| \mathbf{z} - (\tilde{\mathbf{z}} - \nabla f(\tilde{\mathbf{z}})) \|^2$$
$$\text{s.t.} \quad (34), (54). \qquad (57)$$

The remaining details of the derivation are omitted due to space considerations.

VI. SIMULATION RESULTS

In this section, we use Monte Carlo simulations to demonstrate the benefits of the proposed CCCP-based and low complexity IBCD algorithms for RACO systems in terms of the end-to-end delay and system energy consumption. The simulations are run on a desktop computer with (Intel i7-920) CPU running at 2.66 GHz and 24 Gbytes RAM, while the simulation parameters are set as follows unless specified otherwise. All the channel gains are independently generated based on a Rayleigh fading model with an average gain factor of $\sigma_h^2 = E[|h|^2] = 10^{-3}$ [35]. The radio bandwidth available for data transmission from user A to user B via the MERS is $W = 40$ MHz for the combination of the AF and DF schemes. The background noise at MERS and user B is -169 dBm/Hz [36]. The maximum transmit power levels of user A and the MERS are set to $P_A^{\max} = 1$ Watts and $P_R^{\max} = 5$ Watts, respectively. The maximum computation speed of user A and the MERS are characterized by $F_l^{\max} = 800$ MHz and $F_r^{\max} = 2.4$ GHz, respectively [37]. For user A, the data size of the tasks before computation follows a uniform distribution over the interval $[4 \cdot 10^5, 2 \cdot 10^6]$ bits, the conversion ratio is fixed to $\rho = 0.1$, and the required number of CPU cycles per bit for both user A and the MERS is set to $K \triangleq K_l = K_r = 10^3$ cycles/bit [38]. Furthermore, the power consumption coefficients for the given chip architecture are set as $\eta \triangleq \eta_l = \eta_r = 10^{-28}$ (Watts×s³) [38]–[40]. In the implementation of the IBCD algorithm, the smoothness factor β in (53) is set to 10. All results are obtained by averaging over 100 independent channel realizations. For convenience, the simulation parameters are listed in Table I.

TABLE I: Simulation parameters

| Parameters	Value
Radio bandwith of AF and DF subchannels W	40 MHz
Average gain factor σ_h^2	10^{-3}
Maximum transmission power of user A $P_A^{\max}$	1 Watts
Maximum transmission power of the MERS $P_R^{\max}$	5 Watts
Maximum computation capacity of user A $F_l^{\max}$	800 MHz
Maximum computation capacity of the MERS $F_r^{\max}$	2.4 GHz
Size of tasks before computation L	$[4 \cdot 10^5, 2 \cdot 10^6]$ bits
CPU cycles required to process a bit K	1000 cycles/bit
Conversion ratio ρ	0.1
Coefficient depending on chip architecture η	10^{-28} Watts×s³,
Smoothness factor β	10
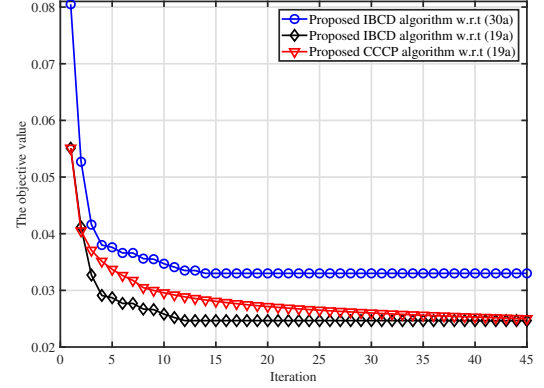
Fig. 3: Convergence behavior for the proposed algorithms for the case $\gamma = 0.01$ (J·sec^{-1}).

TABLE II: Comparison of execution times for CCCP-based and IBCD algorithms.

	Average number of iterations to converge	Average execution time per iteration
IBCD	20	0.1 s
CCCP	45	1.82 s

A. Convergence Performance

We begin by studying the convergence performance of the proposed CCCP-based and low-complexity IBCD algorithms. Without loss of generality, we choose $\gamma = 0.01$ (J·sec^{-1}) for characterizing the convergence performance of the proposed algorithms in Fig. 3 and Table II. For the CCCP-based algorithm, Fig. 3 plots the values of the objective function (19) versus the iteration number, while for the IBCD algorithm, the values of both the objective function (19) and its smoothed approximation (30) versus iteration number are plotted. These curves reveal that despite the existence of a gap between the two objective functions (i.e., green versus red lines), the low-complexity IBCD algorithm monotonically converges to the same value as that achieved by the CCCP-based algorithm, (i.e., green versus blue lines). We note that the IBCD algorithm can achieve faster convergence than the CCCP-based algorithm. In addition, since the CCCP-based algorithm requires solving a sequence of complex convex problems, a single iteration of IBCD runs much faster than the corresponding CCCP iteration, as shown by the average run time data in Table II. Hence, the IBCD algorithm is more efficient than the CCCP-based algorithm.

B. Performance of Proposed Algorithms for General Case: HR Scheme

Let us now investigate the performance of the proposed resource allocation algorithms when applied to the general case of the RACO system with HR architecture. The tradeoff between the system's energy consumption and execution delay for the CCCP-based and IBCD algorithms is illustrated in Fig. 4a for $F_r^{\max} = 1.2, 2.4,$ and 4.8 GHz. It can be observed that the energy consumption increases while the execution delay decreases as the weighting parameter γ in (19) increases. When γ is relatively large, our approach gives more weight to the delay minimization; consequently, our proposed algorithms can achieve the minimum execution delay (i.e., vertical blue dashed line). Conversely, when γ is relatively small, our approach gives more emphasis to the minimization of the energy consumption minimization and, in particular, tends to yield the same energy consumption irrespective of the value of $F_r^{\max}$. This is because the minimum energy consumption is achieved when F_r is very small [cf. (16)]. This reveals a fundamental design principle for RACO systems: when our design emphasis is on the minimization of energy consumption, there is no need to deploy too excessive resources at the MERS. Besides, we also note from Fig. 4a that the performance of the IBCD algorithm is inferior to that of the CCCP-based algorithm, especially when γ is relatively large. In effect, it appears that the smooth approximation of execution delay t_{sys} in (35) leads to a notable performance loss when more weight is given to delay minimization. Even so, the IBCD algorithm is still very promising due to its lower computational complexity as demonstrated earlier.

Fig. 4b shows the values of the objective function (19) versus the distance between user A and the MERS, with $\gamma = 0.5$ and 0.05. In this simulation, we consider a practical scenario where the distance between user A and user B is set to 180 meters(m), with the MERS being located on the line segment joining them at a distance D from user A, where $10 \leq D \leq 170$m. The following path loss model is employed for the wireless links: $\text{PL}_0 \left(\frac{d}{d_0} \right)^{-\chi}$, where d denotes the distance between the transmitter and receiver, $\chi = 3$ is the path loss exponent, and $\text{PL}_0 = -60$ dB denotes the path loss at a reference distance of $d_0 = 10$m [22]. It can be observed that the objective function first decreases to reach a minimum at $D = 90$m and then increases as D further increases. This is due to the fact that with increasing D, the wireless channel between user A and the MERS becomes weaker, while that between the MERS and user B becomes stronger. This benefits the forwarding of computational results from the MERS to user B, but incurs increased energy consumption and transmission delay from user A to the MERS. These observations are consistent with the theoretical analysis in [22], where the optimal location for the relay is shown to be at the middle point between user A and B.

C. Performance of Proposed Algorithms for Special Cases: AF and DF Schemes Only

These special cases occur in the limit $\{\alpha = 0, \nu = 0\}$ and $\{\alpha = 1, \nu = 1\}$: in the former case, user A performs all of
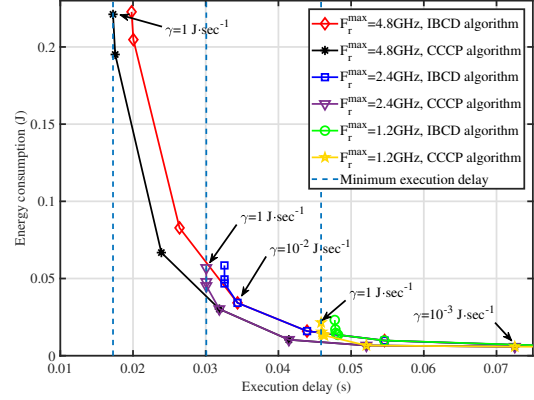
(a)
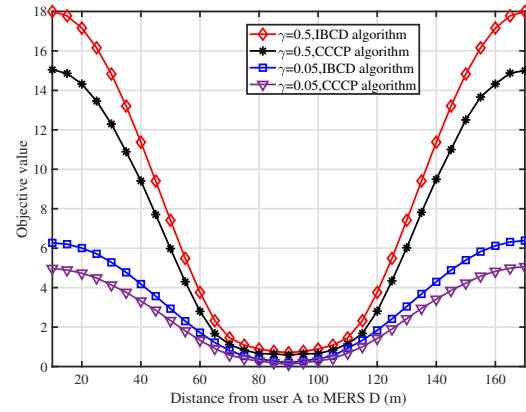
(b)

Fig. 4: (a) Energy consumption versus execution delay for different algorithms; (b) objective function value versus the distance between user A and MERS for different algorithms.

its computational tasks locally and sends the results to user B using the AF relaying scheme; while in the latter case, user A offloads all of its tasks to the MERS for mobile edge execution using the DF scheme. In these two cases, problem **P1** reduces to the simpler forms (51) and (42) respectively, which in turn lead to the more efficient algorithmic solutions developed in Section V, which we now investigate. In addition to the proposed HR architecture, the following baseline schemes are considered for comparison:

1) The proposed HR architecture with time division (TDHR): The locally computed results and the offloaded raw tasks are each transmitted using the complete available radio spectrum but different time slots. The durations of time slots for the AF and the DF scheme are the same. In addition, the locally computed results are transmitted just after the offloaded raw task.

2) The proposed HR architecture with frequency division (FDHR): This scheme employs both the AF and the DF relaying over two orthogonal frequency bands. Moreover, the AF and the DF scheme occupy the same channel bandwidth of $W/2 = 20$ MHz.
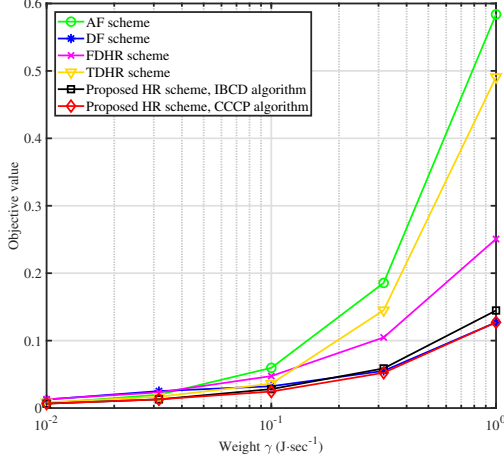
Fig. 5 illustrates the objective function (19) versus the

Fig. 5: Objective function value versus the weight factor between execution delay and energy consumption for different relay schemes.

(a)

(b)

Fig. 6: Objective function value versus (a) the average number of computation bits for $\gamma = 0.01$ (J·sec^{-1}); (b) the average number of computation bits for $\gamma = 1$ (J·sec^{-1}).

weight factor γ between execution delay and energy consumption for different schemes and algorithms. It can be observed that for all schemes under comparison, the value of the objective function increases with γ. First, we discuss the performance comparison between the proposed HR scheme and the AF and the DF only schemes. It is seen that the DF only scheme outperforms the AF scheme for larger γ. This indicates that when more emphasis is given to the minimization of the delay, the performance of the RACO system can be further improved by employing the DF scheme, as compared to the AF scheme. When γ is relatively small and more weight is given to the energy consumption minimization, the AF scheme outperforms its DF counterpart. Furthermore, it is interesting to note that the HR scheme with CCCP-based optimization achieves a lower weighted sum of the execution delay and of the energy consumption than that of both the AF and the DF schemes for all possible values of γ. These results demonstrate the efficiency of the HR architecture in handling different scenarios for user preferences (i.e. weighting factor γ) and its ability to strike a better balance between energy minimization and execution delay, thereby endowing added flexibility to the RACO system. Moreover, it is observed that the performance of the AF scheme is close to that of the proposed HR scheme at relatively smaller γ, while the performance of the DF scheme is enhanced monotonically and coincides with that of the proposed HR scheme when $\gamma = 1$. Secondly, we compare the performance of the HR schemes with different division modes. It should be emphasized that the TDHR scheme tends to be superior at smaller γ but performs worse at larger γ. This is because the TDHR scheme allocates two different time slots for transmitting the locally computed results and the offloaded raw tasks, which is inefficient when more emphasis is given to the minimization of the delay. As for the FDHR scheme, its performance is also inferior due to its fixed and ineffective spectrum utilization method. Finally, it is observed that the proposed HR architecture achieves significant gain over the FDHR and the TDHR schemes for
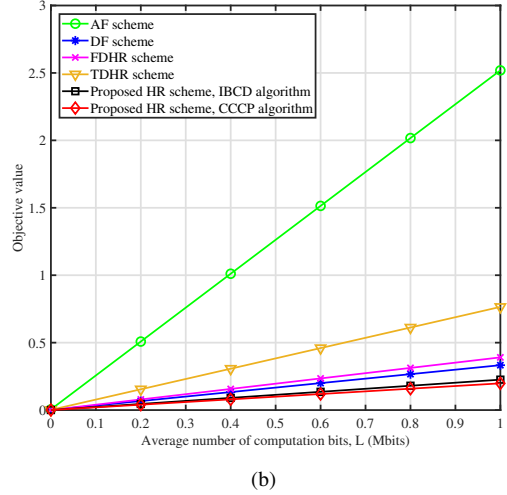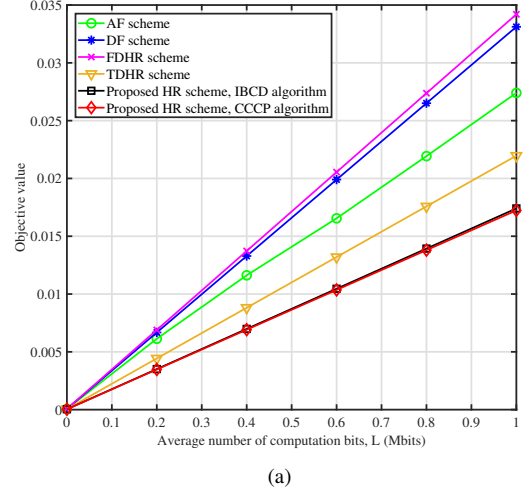
all γ regime, because both the frequency resources and time resources are fully utilized by the proposed HR architecture.

Fig. 6a and Fig. 6b show plots of the objective function (19) versus the average number of computation bits L for different schemes/algorithms and for two different choice of weighting factor, i.e., $\gamma = 0.01$ and $\gamma = 1$, respectively. It can be observed that the objective function values of all schemes and for the two different choices of γ linearly with respect to the number of computation bits L. This is intuitively satisfying since more computation bits pose more stringent requirements on both computation and radio resources, which incurs increased energy consumption and time delay for the RACO system. Note that in this paper, we aim to minimize the weighted sum of execution delay and energy consumption by appropriately allocating both the computational and communication resources. For such a scenario, the tradeoff between execution delay and energy consumption relies more on the inner computational characteristics of tasks, and is

invariant to the input size L. Consequently, the optimum value of the offloading ratio is independent of L. In Fig. 6a, the performance of the AF scheme is superior to that of the DF one, but still not as good as that of the HR architecture. This implies that the AF scheme is more suitable for implementation than the DF scheme when the design emphasis is placed on the minimization of energy consumption. In Fig. 6b the performance of the DF scheme is much better than that of the AF one, and coincides with that of the HR architecture. Hence, the DF scheme is more suitable than the AF scheme when the design emphasis is placed on the minimization of the delay. According to (53), the optimal computational speed of user A depends on the weight factor and the CPU's coefficient, but not on the size of the computational tasks. Hence, it can be inferred that an increase of the weight factor γ will pose more stringent requirements on the computation speed of user A. When the optimal computational speed exceeds the allowed maximum for user A, the execution delay of the RACO system will be severely penalized, thereby leading to a poor performance. This further demonstrates the advantages of cooperative computation offloading between user A and the MERS. It is also interesting to note from Fig. 6 that the HR architecture with CCCP-based optimization outperforms both the TDHR scheme and the FDHR scheme in terms of the weighted sum of execution delay and energy consumption performance, which demonstrates the importance of the bandwidth allocation and relay strategy.

VII. CONCLUSION

This paper has investigated the problem of joint cooperative relaying and computation sharing within a MEC context, where the aim is to minimize the weighted sum of the execution delay and the energy consumption in MEC systems. To support RACO, we proposed a HR architecture which combines the merits of AF and DF relaying to enhance performance. To tackle the challenging optimization problem under consideration, where the design variables are highly coupled through the objective and constraint functions, an efficient CCCP-based algorithm was proposed to jointly optimize the bandwidth allocation, the transmit power level of the source user A and the MERS, the computational resources and the percentage of computational tasks offloaded through the DF relay channel, under constraints on the CPU speed

and the transmission power budgets at user A and M-RES. To address the difficulty arising from the high computational complexity of the CCCP-based algorithm, we proposed a simplified algorithm based on a smoothed approximation along with the IBCD method, that takes advantage of the problem structure to reduce complexity. We then considered two special situations corresponding to limiting cases of the offloading ratio, i.e., AF and DF scheme, and for these proposed efficient solutions. Numerical results showed that the proposed HR architecture can achieve better performance than the AF and the DF scheme, and is particularly well-suited for RACO applications due to its great flexibility along with reduced execution delay and energy consumption. Due to the space limitation, there have been various important issues that have not been addressed in this paper, e.g. the more general case with multiple user, bi-directional communication, and so on.

APPENDIX A
DERIVATION OF EQUIVALENT TRANSFORMATION

First, let us focus on the term E_{sys} in problem **P1**. By introducing auxiliary variables t_A, t_{D1} and t_{D2}, corresponding to the upper bounds of the transmitting time delay in each relay scheme, we can move the associated mathematical expressions to constraints. Similarly, we introduce the auxiliary variables s_1 corresponding to the upper bound of $P_1^A P_1^R$ in (19i). The resultant equivalent optimization problem is then given by

$$\min_{\mathbf{x},\boldsymbol{\phi}} \; E_{\text{sys}} + \gamma t_{sys} \tag{58a}$$

$$\text{s.t.} \quad (19b) - (19h), \tag{58b}$$

$$\frac{2(1-\alpha)\rho L}{(1-\nu)W \log_2\left(1 + \frac{P_1^A P_1^R |h_B^{(1)} h_A^{(1)}|^2}{P_1^R |h_B^{(1)}|^2 \sigma_{R1}^2 + \sigma_{B1}^2}\right)} \leq t_{\text{AF}} \tag{58c}$$

$$\frac{\alpha L}{\nu W \log_2\left(1 + \frac{P_2^A |h_A^{(2)}|^2}{\sigma_{R2}^2}\right)} \leq t_{\text{D1}} \tag{58d}$$

$$\frac{\alpha \rho L}{\nu W \log_2\left(1 + \frac{P_2^R |h_B^{(2)}|^2}{\sigma_{B2}^2}\right)} \leq t_{\text{D2}} \tag{58e}$$

$$P_1^A P_1^R \leq s_1, P_1^R \sigma_{R1}^2 + |h_A^{(1)}|^2 s_1 + P_2^R \leq P_R^{\max}. \tag{58f}$$

In order to transform (58c)-(58e) into a simpler form, we introduce an additional set of auxiliary variables, i.e.,

$$f_2(\mathbf{x},\boldsymbol{\phi}) \triangleq \frac{1}{2}[LK_r\eta_r(\alpha^2 + F_r{}^4) + LK_l\eta_l(\alpha + F_l{}^2)^2 + (P_1^A)^2 + t_A{}^2 + \sigma_{R1}^2((P_1^R)^2 + t_A^2)$$
$$+ |h_A^{(1)}|^2(s_1^2 + t_A^2) + (P_2^A)^2 + t_{D1}{}^2 + (P_2^R)^2 + t_{D2}^2]. \tag{63}$$

$$\hat{f}_2(\mathbf{x},\boldsymbol{\phi};\tilde{\mathbf{x}},\tilde{\boldsymbol{\phi}}) = \frac{1}{2}[LK_r\eta_r(\tilde{\alpha}^2 + \tilde{F}_r{}^4) + LK_l\eta_l(\tilde{\alpha} + \tilde{F}_l{}^2)^2 + LK_r\eta_r\tilde{\alpha}(\alpha - \tilde{\alpha}) + 2LK_r\eta_r\tilde{F}_r{}^3(F_r - \tilde{F}_r) + LK_l\eta_l(\tilde{\alpha} + \tilde{F}_l{}^2)(\alpha - \tilde{\alpha})$$
$$+ 2LK_l\eta_l\tilde{F}_l(\tilde{\alpha} + \tilde{F}_l{}^2)(F_l - \tilde{F}_l)] + (P_1^{\tilde{A}})^2 + \tilde{t}_A^2 + \sigma_{R1}^2((P_1^{\tilde{R}})^2 + \tilde{t}_A^2) + |h_A^{(1)}|^2(\tilde{s}_1^2 + \tilde{t}_A^2) + (P_2^{\tilde{A}})^2 + \tilde{t}_{D1}^2 + (P_2^{\tilde{R}})^2$$
$$+ \tilde{t}_{D2}^2] + [\tilde{P}_1^A(P_1^A - \tilde{P}_1^A) + \tilde{t}_A(t_A - \tilde{t}_A) + \sigma_{R1}^2(\tilde{P}_1^R(P_1^R - \tilde{P}_1^R) + \tilde{t}_A(t_A - \tilde{t}_A)) + |h_A^{(1)}|^2(\tilde{s}_1(s_1 - \tilde{s}_1)$$
$$+ \tilde{t}_A(t_A - \tilde{t}_A)) + \tilde{P}_2^A(P_2^A - \tilde{P}_2^A) + \tilde{t}_{D1}(t_{D1} - \tilde{t}_{D1}) + \tilde{P}_2^R(P_2^R - \tilde{P}_2^R) + \tilde{t}_{D2}(t_{D2} - \tilde{t}_{D2})]. \tag{64}$$

$\{R_A, R_{D1}, R_{D2}, \Lambda_A, \Lambda_{D1}, \Lambda_{D2}, \varphi_1, \varphi_2, \varphi_3, s_2\}$. Here, we focus on the required manipulations for (58c) as an example:

$$R_A \le (1-\nu)W\Lambda_A/2 \le (1-\nu)W\log_2(1+1/\varphi_1)/2,$$

$$\frac{1}{\varphi_1} \le \frac{s_2|h_B^{(1)}h_A^{(1)}|^2}{P_1^R|h_B^{(1)}|^2\sigma_{R1}^2 + \sigma_{B1}^2} \le \frac{P_1^A P_1^R|h_B^{(1)}h_A^{(1)}|^2}{P_1^R|h_B^{(1)}|^2\sigma_{R1}^2 + \sigma_{B1}^2}.$$

It should be emphasized that the auxiliary variable s_2 is introduced as the lower bound of $P_1^A P_1^R$. Then, (58) can be formulated as the equivalent problem below:

$$\min_{\mathbf{x},\boldsymbol{\phi}} E_{\text{sys}} + \gamma t_{\text{sys}} \tag{59a}$$

$$\text{s.t.} \quad (1-\alpha)\rho L \le t_A R_A, 2R_A \le (1-\nu)W\Lambda_A, \tag{59b}$$

$$\Lambda_A \le \log_2(1+1/\varphi_1), \alpha L \le t_{D1}R_{D1}, \tag{59c}$$

$$R_{D1} \le \nu W\Lambda_{D1}, \ \Lambda_{D1} \le \log_2(1+1/\varphi_2), \tag{59d}$$

$$\alpha\rho L \le R_{D2}t_{D2}, \ R_{D2} \le \nu W\Lambda_{D2}, \tag{59e}$$

$$\Lambda_{D2} \le \log_2(1+1/\varphi_3), \sigma_{R2}^2 - \varphi_2 P_2^A|h_A^{(2)}|^2 \le 0, \tag{59f}$$

$$|h_B^{(1)}|^2\sigma_{R1}^2 P_1^R + \sigma_{B1}^2 - |h_B^{(1)}h_A^{(1)}|^2\varphi_1 s_2 \le 0, \tag{59g}$$

$$|h_B^{(1)}|^2\sigma_{R1}^2 P_1^R + \sigma_{B1}^2 - |h_B^{(1)}h_A^{(1)}|^2\varphi_1 s_2 \le 0, \tag{59h}$$

$$\sigma_{B2}^2 - \varphi_3 P_2^R|h_B^{(2)}|^2 \le 0, s_2 \le P_1^A P_1^R \le s_1, \tag{59i}$$

$$(19b) - (19h). \tag{59j}$$

Finally, we must address the difficulty posed by the term t_{sys} in (58). To this end, we can move the term t_{sys} into the constraints and introduce the auxiliary variables $\{t_s, t_l, t_r\}$, now corresponding to the upper bound of the whole execution time, local execution time, and edge execution time respectively, which yields equivalent yet more tractable constraints as follows,

$$t_l + t_A \le t_s, \ t_{D1} + t_r + t_{D2} \le t_s \tag{60}$$

$$t_l \ge K_l(1-\alpha)L/F_l, \ t_r \ge K_r\alpha L/F_r. \tag{61}$$

Substituting (60) and (61) in problem (59), we obtain the equivalent problem **P2**. Complete this proof.

APPENDIX B
DERIVATION OF (23)

We note that $f_3(\mathbf{x},\boldsymbol{\phi})$ is nonconvex due to the products of optimization variables. To tackle this nonconvexity by applying the CCCP, we first need to transform $f_3(\mathbf{x},\boldsymbol{\phi})$ into a difference-of-convex (DC) program. Focusing on the operation of the last term $P_2^R t_{D2}$ in $f_3(\mathbf{x},\boldsymbol{\phi})$ as an example, we have:

$$P_2^R t_{DF2} = \frac{1}{2}[(P_2^R + t_{D2})^2 - (P_2^R)^2 - t_{D2}^2] \tag{62}$$

Algorithm 3: Proposed bisection algorithm for the problem (38)

0. Input: the current point $\tilde{P}_1^A, \tilde{P}_2^A, \tilde{P}_1^R$ and $\tilde{P}_2^R$.
1. Define the lower and upper bounds of the Lagrange multiplier respectively λ_u and λ_l, and the tolerance of accuracy ϵ. Initialize the algorithm with $0 \le \lambda_l < \lambda_u$.
2. **Repeat**
3.   Let $\lambda \leftarrow \frac{\lambda_l + \lambda_u}{2}$.
4.   Update $\overline{P}_1^R$ and $\overline{P}_2^R$ according to (65) and (66), respectively.
5.   **If** point (C_1, C_2) is inside of the feasible region,
         update $\overline{P}_1^A = C_1$ and $\overline{P}_1^A = C_2$, respectively;
6.   **else** update $\overline{P}_1^A$ and $\overline{P}_2^A$ according to (71) and (72), respectively.
7.   **If** $U(\overline{\mathbf{y}}; \tilde{\mathbf{y}}) \ge 0$, let $\lambda_l \leftarrow \lambda$; **else** let $\lambda_u \leftarrow \lambda$.
8. **Until** $| \lambda_u - \lambda_l | \le \epsilon$.

We can follow the same approach to handle the remaining nonconvex terms in $f_3(\mathbf{x},\boldsymbol{\phi})$, and finally obtain $f_3(\mathbf{x},\boldsymbol{\phi}) \triangleq f_1(\mathbf{x},\boldsymbol{\phi}) - f_2(\mathbf{x},\boldsymbol{\phi})$, where $f_1(\mathbf{x},\boldsymbol{\phi})$ and $f_2(\mathbf{x},\boldsymbol{\phi})$ are respectively defined in (24) and (63) as displayed at the bottom of this page.

Based on the CCCP concept [21], we approximate the convex function $f_2(\mathbf{x},\boldsymbol{\phi})$ in the ith iteration by its first order Taylor expansion around the current point $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}})$ in (64) as displayed at the bottom of this page.

Therefore, using the above results we can obtain a locally tight upper bound for the objective function of problem **P2**, i.e., $f_3(\mathbf{x},\boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}}) \triangleq f_1(\mathbf{x},\boldsymbol{\phi}) - \hat{f}_2(\mathbf{x},\boldsymbol{\phi}; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\phi}})$.

APPENDIX C
SOLVING PROBLEM (41) FOR UPDATING $\overline{y}$

In this part, we derive each step of the update procedure for solving (41).

1) Subproblem w.r.t $\{P_1^R, P_2^R\}$: By applying the first-order optimality condition, we obtain a closed-form solution as follows,

$$\overline{P}_1^R = \min(\max(0, D_1 + D_2 P_1^A), P_R^{\max}), \tag{65}$$

$$\overline{P}_2^R = \min(\max(0, \tilde{P}_2^R - \partial f_\beta(\tilde{\mathbf{y}})/\partial P_2^R - \lambda), P_R^{\max}), \tag{66}$$

where both D_1 and D_2 are defined (67) in as displayed at the bottom of the next page.

2) Subproblem w.r.t $\{P_1^A, P_2^A\}$: Substituting (65) and (66) into (41), we can obtain a quadratic optimization problem w.r.t $\{P_1^A, P_2^A\}$ as follows

$$\min_{P_1^A, P_2^A} A_1(P_1^A - C_1)^2 + (P_2^A - C_2)^2 \tag{68a}$$

$$\text{s.t.} \quad (19e), (19g) \tag{68b}$$

$$D_1 \triangleq \left(1 + \lambda|h_A^{(2)}|^2\right)^{-1}(\tilde{P}_1^R - \partial f_\beta(\tilde{\mathbf{y}})/\partial P_1^R - \lambda\sigma_{R1}^2 + \lambda|h_A^{(1)}|^2\tilde{P}_1^R), \quad D_2 \triangleq -\left(1 + \lambda|h_A^{(2)}|^2\right)^{-1}\lambda|h_A^{(1)}|^2. \tag{67}$$

$$A_1 \triangleq 1 + D_2^2 + \lambda|h_A^{(1)}|^2(1+D_2)^2, \quad C_2 \triangleq \tilde{P}_2^A - \partial f_\beta(\tilde{\mathbf{y}})/\partial P_2^A,$$

$$C_1 \triangleq \frac{\tilde{P}_1^A - \frac{\partial f_\beta(\tilde{\mathbf{y}})}{\partial P_1^A} + D_2(\tilde{P}_1^R - \frac{\partial f_\beta(\tilde{\mathbf{y}})}{\partial P_1^R} - D_1 - \lambda\sigma_{R1}^2) + \lambda|h_A^{(1)}|^2(D_2\tilde{P}_1^R + \tilde{P}_1^A - (1+D_2)D_1)}{1 + D_2^2 + \lambda|h_A^{(1)}|^2(1+D_2)^2}. \tag{69}$$

where all the term A_1, C_1, and C_2 are defined in (69) in as displayed at the bottom of the next page. Solving problem (68) is equivalent to computing a projection of the point (C_1, C_2) onto the set $\Omega_1 = \{(P_1^A, P_2^A) \mid P_1^A + P_2^A \leq P_A^{\max}, P_1^A \geq 0, P_2^A \geq 0\}$. When the point (C_1, C_2) is inside of the feasible region, the optimal solution to problem (68) is immediately obtained as $\overline{P}_1^A = C_1, \overline{P}_2^A = C_2$. When the point (C_1, C_2) is not inside of the feasible region, the optimal solution to (68) can be obtained on the boundary of the feasible region. In other words, the constraint (19e) is strictly satisfied, i.e., $P_2^A = P_{\max}^A - P_1^A$. By substituting it into (68), we can obtain a quadratic formulation w.r.t P_1^A:

$$\min_{\{0 \leq P_1^A \leq P_A^{\max}\}} A_1(P_1^A - C_1)^2 + (P_A^{\max} - C_2 - P_1^A)^2 \quad (70)$$

Applying the first-order optimality condition yields a closed-form solution as follows

$$\overline{P}_1^A = \min(\max(0, (A_1 + 1)^{-1}(A_1 C_1 + P_A^{\max} - C_2)), P_A^{\max}). \quad (71)$$

Given the optimal solution $\overline{P}_1^A$ in (70), the optimal $\overline{P}_2^A$ can be expressed as

$$\overline{P}_2^A = P_A^{\max} - \overline{P}_1^A. \quad (72)$$

Furthermore, since the objective function of problem (41) given $\lambda > 0$ is strictly convex, problem (41) has a unique solution. It follows that $h(\lambda)$ is differentiable for $\lambda > 0$ and its derivative is $U(\overline{\mathbf{y}}; \tilde{\mathbf{y}})$. Consequently, the dual problem (40) can be efficiently solved using the bisection method, which is summarized in Algorithm 3.

REFERENCES

[1] CISCO, "Cisco Visual Networking Index: Forecast and Trends, 2017C2022 white paper," [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html, 2018.

[2] European Telecommunications Standards Institute (ETSI), "Mobile-edge-computing-Introductory technical white paper," Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge-computing introductory technical white paper v1.

[3] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, "A survey of computation offloading for mobile systems", *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129-140, 2013.

[4] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.

[5] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July. 2016, pp. 1451-1455.

[6] X. Chen et al., "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE Trans. Netw.*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.

[7] C. You, K. Huang, H. Chae, B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading", *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397-1411, Mar. 2017.

[8] 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.

[9] Z. Jiang, and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices", *IEEE Access*, vol. 3., pp.2306-2316, 2015.

[10] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf.*, Washington, Dec. 2016, pp. 1-6.

[11] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Info. Process. Over Networks*, vol. 1, no. 2, pp.89-103, Jun. 2015.

[12] C. Wang, C. Liang, F. Richard Yu, Q. Chen and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924-4938, Aug. 2017.

[13] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268-4282, Oct. 2016.

[14] S. Barbarossa, E. Ceci, M. Merluzzi and E. Calvanese, "Enabling effective mobile edge computing using millimeter wave links," in *Proc. IEEE Int. Communication Conference*, Paris, May 21, 2017.

[15] L. Wang, and L. Hanzo, "Dispensing with channel estimation: differentially modulated cooperative wireless communications," *IEEE Commun. Surveys and Tutorials*, vol. 14, no. 3, pp. 836-857, Sep. 2011.

[16] S. Wang, and S. Dey, "Model and characterizing user experience in a cloud server based mobile gaming approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1-7.

[17] J. Ren et al., "An edge-computing based architecture for mobile augmented reality," *IEEE Network*, vol. 33, no.4, Jul./Aug. 2019.

[18] Y. Ai, L. Wang, Z. Han, P. Zhang, and L. Hanzo, "Social networking and caching aided collaborative computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 149-155, Dec. 2018.

[19] W. Cai, R. Shea, C. Huang, K. Chen, J. Liu, V. C. M. Leung, and C. Hsu, "A survey on cloud gaming: Future of computer games", *IEEE Access*, vol. 4, pp. 7605-7620, Aug. 2016.

[20] H. Jin, X. Zhu, and C. Zhao, "Computation offloading optimization based probabilistic SFC for mobile online gaming in hetergenous network", *IEEE Access*, vol. 7, pp. 52168-52180, Apr. 2019.

[21] A. L. Yuille and A. Rangarajan, "The concave-convex procedure", *Neural Comput.*, vol. 15, pp. 915-936, Apr. 2003.

[22] C. Cao, F. Wang, J. Xu, R. Zhang and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, Jun. 2018.

[23] N. J. Laneman et al., "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory,* vol. 50, pp. 3062-3080, Dec. 2004.

[24] B. Rankov and A. Wittneben, "Spectral efficiency protocols for halfduplex fading relay channels," *IEEE J. Sel. Areas Commun.*, 2007, 525, (2), pp. 379-389.

[25] N. Zlatanov, V. Jamali, and R. Schober, "Achievable rates for the fading half-duplex single relay selection network using buffer-aided relaying," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4494-4507, Aug. 2015.

[26] *Technical Specification Group Radio Access Network: Further Advancements for E-UTRA Physical Layer Aspects,* 3GPP TR 36.814. [Online]. Available: http://www.3gpp.org.

[27] G. Scutari, F. Facchinei, L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization-part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929-1944, Apr. 2017.

[28] CVX Research, Inc. (Sep. 2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1.* [Online]. Available: http://cvxr.com/cvx.

[29] Lanckriet, Gert R., and Bharath K. Sriperumbudur, "On the convergence of the concave-convex procedure," *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp.1759-1767.

[30] Y. Cheng and M. Pesavento, "Joint optimization of source power allocation and distributed relay beamforming in multiuser peer-to-peer relay networks," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2962-2973, Jun. 2012.

[31] J. Tranter, N. D. Sidiropoulos, X. Fu, and A. Swami, "Fast unit-modulus least squares with applications in beamforming," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2875-2887, Jun. 2017.

[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183-202, 2009.

[33] M. Shao, Q. Li, W.K. Ma, A. Man-Cho, "Minimum symbol error rate-based constant envelope precoding for multiuser massive MISO downlink," to appear in the *Proc.* of the 2018 *IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

[34] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[35] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, Mar. 2017.

[36] Y. Yu et al., "Joint subcarrier and CPU time allocation for mobile edge computing", *Proc. IEEE Global Commun. Conf. (Globecom)*, Dec. 2016, pp. 1-7.

[37] O. Munoz et al., "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738-4755, Oct. 2015.

[38] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-User mobile edge computing", *Proc. IEEE Global Commun. Conf. (Globecom)*, Dec. 2017, pp. 1-7.

[39] X. Chen, Q. Shi, Y. Cai, and M. Zhao, "Joint cooperative computation and interactive communication for relay-assisted mobile edge computing," in *Proc. IEEE VTC Fall*, Chicago, USA, Aug. 2018.

[40] X. Chen, Y. Cai, M. Zhao, and M. Zhao, "Joint computation offloading and resource allocation for min-max fairness in MEC systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakech, Morocco, Apr. 2019.

[41] Q. Shi, H. Sun S. Lu, M. Hong, and M. Razaviyayn, "Inexact block coordinate descent methods for symmetric nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5995-6008, Nov. 2017.