Delay-Optimal Coded Offloading for Distributed Edge Computing in Fading Environments

Xiaofan He, *Senior Member, IEEE*, Tianheng Li, *Student Member, IEEE*, Richeng Jin, *Member, IEEE*, and Huaiyu Dai, *Fellow, IEEE*

***Abstract*— The rapid growth in scale and complexity of mobile applications fosters the development of the coded edge computing paradigm. By exploiting the redundancy in the encoded subtasks, coded edge computing enables collaborative transmission of multiple edge nodes and is promising for distributed computing in wireless fading environments. Nonetheless, to the best of our knowledge, due to challenges arising from the selection of the coding parameters, offloading strategy design for coded edge computing in general fading environments still remains open. With this consideration, the coded offloading problem is studied in this work and a delay-optimal coded offloading scheme is proposed. In particular, when the offloaded tasks are encoded by (k, r) linear codes, transmission diversity gains can be obtained by performing edge node selection to mitigate fading. However, the corresponding optimization problem turns out to be a highly non-trivial non-linear mixed-integer programming. To this end, through in-depth analysis based on order statistics, it is found that the average processing delay of the offloaded tasks admits a favorable V-structure with respect to the coding parameter r, under arbitrary fading distribution. This key theoretic result allows us to efficiently solve the original problem using monotonic optimization. Simulations are conducted to validate our analysis and corroborate the effectiveness of the proposed scheme.**

***Index Terms*— Mobile edge computing, coded computing, task offloading.**

I. Introduction

THE recently advocated edge computing [1] is expected to remarkably improve the computing experience of mobile users. However, the surge of advanced and sophisticated mobile applications such as artificial intelligence, blockchain, and big data is pushing the limits of edge computing. When facing these large-scale and intensive computation tasks, a single edge node likely becomes overwhelmed. On the other hand, riding on the wave of technological development of semiconductor, many electronic devices and infrastructures are equipped with competent computing power and submerge mobile users with a distributed computing environment teeming with potential edge nodes. These ignite the research interests in distributed edge computing techniques that can leverage the dispersed processing capabilities of multiple computing devices for enhanced performance [2]–[4].

Inspired by the idea of erasure codes [5], coded edge computing [3], [4], [6]–[10] was recently proposed as a promising distributed edge computing paradigm. In coded edge computing, the original task will be encoded into multiple smaller subtasks and offloaded to different edge nodes. Due to the redundancy among the encoded subtasks, the mobile device can recover the computation results by receiving feedbacks from only a subset of edge nodes, thereby effectively mitigating straggling in the conventional distributed computing [11]. Besides, coded edge computing also brings bountiful opportunities to reduce transmission delay caused by wireless fading in distributed computing. In particular, the rich redundancy in the encoded subtasks enables collaborative transmission among the edge nodes to mitigate fading. In the literature, collaborative communication techniques, such as zero-forcing precoding, beamforming, and interference alignment, have already been considered to reduce the downlink transmission delay in coded edge computing [6], [12]–[14]. In view of these appealing features, substantial research efforts have been devoted to coded edge computing and the associated workload allocation [15], [16], straggler exploitation [17], [18], non-linear computation [19], as well as coded machine learning [20] issues.

Nonetheless, coded edge computing entails new challenges to task offloading [1], [21], [22]. Specifically, as task encoding leads to remarkable changes of computation and communication, the selection of the coding parameters becomes vital for offloading scheme design. Although several pioneering works [6], [12]–[14] have studied the (asymptotic) communication-computation tradeoff under different coding parameters, most of them only consider the high signal-to-noise ratio (SNR) scenarios. The influence of the coding parameters on task processing time in wireless environments with general SNR warrants further investigation. Besides, most of the existing works either assume a fixed choice of the coding parameters or employ brute-force search to find the optimal coding parameters [14], [23]–[25]. Although there

Manuscript received 30 August 2021; revised 23 January 2022 and 3 May 2022; accepted 25 June 2022. Date of publication 13 July 2022; date of current version 12 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61901305, in part by the Yellow Crane Talents Program under Grant 1501-230100036, in part by the Fundamental Research Funds for the Central Universities under Grant 2042021kf0017, in part by the WHU-DKU Collaborative Research Seed under Grant WHUDKUZZJJ202207, in part by the Wuhan University Start-up under Grant 1501600460001, and in part by the US National Science Foundation under Grant CNS-1824518. The associate editor coordinating the review of this article and approving it for publication was X. Gong. *(Corresponding author: Xiaofan He.)*

Xiaofan He and Tianheng Li are with the School of Electronic Information, Wuhan University, Wuhan 430000, China (e-mail: xiaofanhe@whu.edu.cn; tianhengli@whu.edu.cn).

Richeng Jin was with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA. He is now with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310058, China (e-mail: rjin2@ncsu.edu; richengjin@zju.edu.cn).

Huaiyu Dai is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA (e-mail: hdai@ncsu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3187427.

Digital Object Identifier 10.1109/TWC.2022.3187427

have already been some recent pioneering efforts on this aspect, only approximate optimal coding parameter selection strategy is obtained for the special case of exponentially distributed transmission time [26]. An efficient method for finding the optimal coding parameters under *arbitrary* fading distributions is yet to be developed. Due to these difficulties, to the best of our knowledge, the problem of coded task offloading in general fading environments still remains open.

With the above consideration, the coded offloading problem, which involves a joint design of the communication parameter, offloading ratio, and the coding parameters, is studied in this work, and a delay-optimal coded offloading scheme is proposed. Specifically, in the proposed scheme, when tasks are encoded by (k, r) linear codes, transmission diversity gains can be obtained by performing edge node selection to mitigate fading. However, the corresponding optimization problem turns out to be a highly non-trivial non-linear mixed-integer programming. To this end, through in-depth analysis based on order statistics [27], it is found that the average processing delay of the offloaded tasks admits a favorable V-structure with respect to (w.r.t.) the coding parameter r under arbitrary fading distributions. This key theoretic result allows us to efficiently solve the original problem using monotonic optimization [28]. The main contributions of this work are summarized as follows:

- To the best of our knowledge, this work is among the first to consider the coded offloading problem in general fading environments;
- It is proved that the average task processing delay admits a V-structure w.r.t. the coding parameter r under *arbitrary* fading distributions;
- An efficient algorithm for finding the delay-optimal coded offloading strategy is developed.

The rest of this paper is organized as follows. Related works are discussed in Section II. System model and problem formulation are presented in Section III. Several important properties of the optimal coding parameters are established in Section IV. The proposed delay-optimal coded offloading algorithm is developed in Section V, and the simulation results are given in Section VI. Finally, conclusions and future works are discussed in Section VII.

II. Related Works

The essential idea of coded edge computing may be traced back to prior research on classical coded storage and caching and existing research in this area may be categorized into two directions [4]. One line of works mainly aim to reduce the amount of data exchange among distributed computing nodes in the MapReduce framework through task encoding and multicast [29]–[31]. However, these works generally neglect the communications between the mobile users and the computing nodes and thus are different from the coded offloading issue considered in our work. The second line of works are more related and they mainly target addressing the straggling issue in distributed computing [4] via ingenious task encoding as discussed below.

In literature, existing works on coded computing with stragglers mainly focused on addressing the issues of delay analysis [24]–[26], dynamic task encoding [7], [32], coded computing of nonlinear tasks [19], as well as straggler exploitation [17], [18], [33]. Specifically, the average task processing delays under different task encoding mechanisms are analyzed in [24]. By assuming exponentially distributed edge computing time and packet transmission time, the lower and the upper latency bounds are derived for coded computing with packet retransmission [26]. The quantitative relations between the task encoding redundancy and the straggling distribution are studied in [25]. To handle the frequent leaving and joining of computing nodes in dynamic distributed environments, the notion of elastic computing is developed in [32] based on the coded computing framework. In [7], a fountain code based coded computing scheme is developed for high-mobility edge computing scenarios. To handle nonlinear computation tasks, a novel deep learning based coded computing framework is developed in [19]. Smart coding mechanisms are proposed in [3], [17], [18] and [33] to recycle the partial computing results from straggling nodes. For example, a coded computing scheme based on quantization and sphere decoding is developed in [3] to exploit partially finished computations from straggling edge devices. Nonetheless, these pioneering works either only consider the traditional fading-free wired networks or abstract the wireless channels into simple bit pipes and ideal erasure channels.

Besides mitigating computation straggling, coding can be utilized in edge computing networks to improve communication performance by bringing robustness against channel fading and co-channel interference. For example, by exploiting the broadcast nature of wireless transmission, a novel coded computing scheme is developed in [8] to reduce the uplink transmission delay of mobile edge computing. Besides, advanced cooperative communication techniques have been explored to improve the transmission delay of the computation results in literature. For example, zero-forcing precoding based cooperative transmission has been integrated with coded edge computing in [12] and [34] to reduce the transmission latency between the edge nodes and the users. In [6] and [13], universal coded edge computing schemes are developed to linearly encode data from multiple users, and then zero-forcing precoding cooperative transmission is adopted to eliminate communication interference among the users. Other cooperative transmission methods such as interference alignment and beamforming are also shown beneficial for improving the degree-of-freedom in the downlink transmissions of coded edge computing [14], [35]. However, these pioneering works generally make the ideal assumption of high SNRs. Cooperative transmission based coded computing in general wireless environments and the corresponding coding and transmission parameter optimization still remain underexplored. In addition, most of these works consider cooperative transmission only when the computation results of multiple edge nodes are exactly the same (i.e., repetition coded computation). However, under more general task encoding mechanisms, although the computation results of the edge nodes are not necessarily identical, there is still a high degree of information redundancy. How to carry out cooperative transmission in such cases deserves further studies.

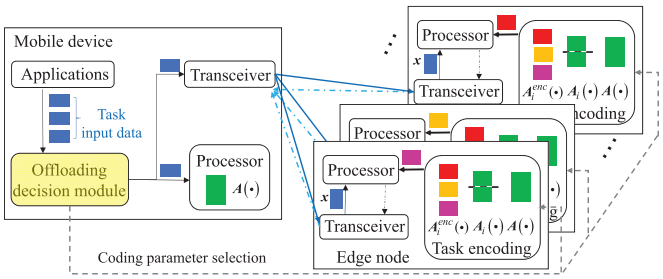


Fig. 1. Coded edge computing.

In contrast to these pioneering works, our work considers general wireless fading environments and leverages node-selection based cooperative transmission to exploit the information redundancy introduced by coded computing to reduce the transmission delay of task offloading.

III. System Model and Problem Formulation

Consider the coded edge computing scenario depicted in Fig. 1, where the mobile device is allowed to offload its tasks to multiple edge servers for distributed processing [3], [7], [8].[1] The mobile device continuously generates a sequence of tasks, and the sequence length m is assumed large. Each task $\mathbf{A}(\mathbf{x})$ consists of input data $\mathbf{x}$ of s bits and a computing function $\mathbf{A}(\cdot)$. The most commonly considered computing function in pertinent literature is matrix multiplication. A comprehensive list of applicable computing functions can be found in [4]. To handle these tasks, the mobile device can simultaneously activate its computing module and transmission module. Specifically, $(1-\rho)m$ tasks will be processed by the local processor and the rest ρm tasks will be offloaded to edge nodes. To capture the fading effect, the wireless channel gains between the mobile device and the edge nodes are assumed independent random variables.

When coded edge computing is adopted, the edge nodes can collaboratively handle one task at a time. In particular, each offloaded task will first be partitioned into r subtasks $\{\mathbf{A}_i(\mathbf{x})\}_{i=1}^r$ with equal computational complexity and size. Then, these subtasks will be transformed into $k\,(\geq r)$ encoded subtasks $\{\mathbf{A}_i^{\text{enc}}(\mathbf{x})\}_{i=1}^k$, by using (k,r) linear code [4], [6], [11], [14], [25].[2] These k encoded subtasks will be processed by k different edge nodes, respectively. It is assumed that the computing functions $\{\mathbf{A}_i^{\text{enc}}(\cdot)\}_{i=1}^k$ of the encoded subtasks are pre-stored on the edge nodes to reduce the uplink transmission cost [14].[3] Hence, the mobile device only needs to transmit the input data $\mathbf{x}$ when offloading a task. Also, it is assumed that the computational complexity and the size of an encoded subtask equals those of an uncoded subtask.

Based on the above model, the processing time $T_{k,r}^o$ of an offloaded task is given by

$$T_{k,r}^o = T_{k,r}^u + T_{k,r}^e + T_{k,r}^d + \tau_0, \tag{1}$$

where $T_{k,r}^u$, $T_{k,r}^e$, and $T_{k,r}^d$ represent the corresponding times of uplink transmission, edge computing, and downlink transmission, respectively; τ_0 represents an upper bound of the task decoding time. The average processing time of a task can be well estimated as follows

$$\begin{aligned}\bar{T}_{k,r} &\triangleq \lim_{m\to\infty}\frac{1}{m}\max\left\{(1-\rho)mT^l, \sum_{j=1}^{m\rho} T_{k,r}^o[j]\right\}\\ &= \max\left\{(1-\rho)T^l, \rho\mathbb{E}\{T_{k,r}^o\}\right\},\end{aligned} \tag{2}$$

where the $\max\{\cdot,\cdot\}$ operator is due to the fact that the local processor and the edge nodes can work in parallel;[4] the second equality follows from the law of large numbers; T^l represents the time of processing a task locally; $T_{k,r}^o[j]$ represents the processing time of the jth offloaded task. Note that the expectation in (2) is over the randomness due to fading of wireless channels.

A. Computation Model

1) Local Computing: The time of computing a task at the local processor of the mobile device can be modelled as follows [1]

$$T_{k,r}^l = \frac{s\beta}{f_{\text{cpu}}^l} = \frac{s\beta}{\min\left\{f_{\max}^l, (p^l/\kappa)^{1/3}\right\}}, \tag{3}$$

where β is the number of required CPU cycles for processing one bit of input data; f_{cpu}^l ($f_{\max}^l$) is the (maximum) frequency of the local processor; p^l is the local computing power of the mobile device; κ is a coefficient depending on the local processor architecture.

2) Edge Computing: When (k,r) encoding is adopted, the computation time of an encoded subtask at edge node i can be modelled as follows [25], [38]–[40]

$$T_{k,r,i}^e = \frac{s\beta}{rf_{\text{cpu}}^e} + \frac{V_i}{\lambda}, \tag{4}$$

where f_{cpu}^e is the (maximum) processor frequency of the edge nodes. The second term of (4) is used to capture the straggling effect, where $\{V_i\}_{i=1}^k$ are independent and identically

[1]When multiple users exist, every user is allowed to adopt this coded offloading procedure and orthogonal channeling can be employed to avoid interference among different users. Note that the interference issue is common to multi-user edge computing/communications and is beyond the scope of this work. Here, our focus is on the single user cases and the in-depth study of its multi-user counterpart is deferred to future works.

[2]Take a matrix multiplication task $\mathbf{Ax}$ consisting of an input vector $\mathbf{x}$ and a computing function $\mathbf{A}$ as an example. When $(3,2)$ code is employed, the two partitioned subtasks will be $\mathbf{A}_1\mathbf{x}$ and $\mathbf{A}_2\mathbf{x}$, where $\mathbf{A}=[\mathbf{A}_1^T,\mathbf{A}_2^T]^T$. The corresponding encoded subtasks can be $\mathbf{A}_1^{\text{enc}}\mathbf{x}=\mathbf{A}_1\mathbf{x}$, $\mathbf{A}_2^{\text{enc}}\mathbf{x}=\mathbf{A}_2\mathbf{x}$, and $\mathbf{A}_3^{\text{enc}}\mathbf{x}=(\mathbf{A}_1+\mathbf{A}_2)\,\mathbf{x}$. In this case, the mobile device only needs to receive the results of any two encoded subtasks to recover that of the original task. For example, when $\mathbf{A}_2^{\text{enc}}\mathbf{x}$ is not received, the mobile device can still recover $\mathbf{Ax}$ from $\mathbf{A}_1^{\text{enc}}\mathbf{x}$ and $\mathbf{A}_3^{\text{enc}}\mathbf{x}-\mathbf{A}_1^{\text{enc}}\mathbf{x}$.

[3]Note that the input data $\mathbf{x}$'s and the computing functions $\mathbf{A}(\cdot)$'s are not necessarily identical for all the tasks. In this case, all the encoded computing functions of these different tasks are assumed to be pre-stored at the edge nodes. With a slight abuse of notation, the corresponding subscripts of task index are omitted for ease of notation.

[4]One may think of dividing these m tasks into two queues. The $(1-\rho)\cdot m$ tasks in the first queue are processed by the local CPU and the rest $\rho\cdot m$ tasks are offloaded to the ENs. Note that this partition is reasonable, as these tasks are assumed independent in our work. Although the tasks in each queue are processed one by one in series order, the processing of these two queues is in parallel, by simultaneously activating the local computing unit and the transmission unit of the mobile device. Similar assumptions have been taken in existing literature [21], [36], [37].

distributed (i.i.d.) random variables with unit mean and λ is a scaling factor. The time of waiting all the k edge nodes to complete their computation is given by

$$T^e_{k,r} = \max_{i=1,\ldots,k} T^e_{k,r,i}. \tag{5}$$

B. Communication Model

1) Uplink Transmission: In the uplink transmission phase, the mobile device broadcasts the input data $\mathbf{x}$ to the edge nodes. Consequently, the uplink transmission time $T^u_{k,r,i}$ to edge node i is given by

$$T^u_{k,r,i} = \frac{s}{B^u \ln\left(1 + \frac{p^t}{B^u N_0} \cdot G^u_i\right)}, \tag{6}$$

where B^u is the uplink bandwidth assumed to be the same for all edge nodes; p^t is the transmit power of the mobile device; N_0 is the noise power spectral density; G^u_i is the random power gain of the uplink channel to edge node i.[5] When k edge nodes are involved, the uplink transmission time $T^u_{k,r}$ is determined by the slowest link and hence is given by

$$\begin{aligned} T^u_{k,r} &= \min\left\{t^u_0, \max_{i=1,\ldots,k} T^u_{k,r,i}\right\} \\ &= \min\left\{t^u_0, \frac{s}{B^u \ln\left(1 + \frac{p^t}{B^u N_0} \cdot G^u_{(1)}\right)}\right\}. \end{aligned} \tag{7}$$

In (7), $G^u_{(1)} \triangleq \min_{i=1,\ldots,k} G^u_i$ represents the smallest one among the power gains of the k uplink channels. To avoid waiting for an indefinitely long time in the uplink transmission phase due to deep fading, the maximum allowable time for the uplink phase is set to t^u_0. When the required transmission time $\max_{i=1,\ldots,k} T^u_{k,r,i}$ exceeds t^u_0, an uplink outage occurs and the corresponding outage probability ϵ^u_{out} is given by

$$\epsilon^u_{\text{out}} = Pr\left\{t^u_0 \le \frac{s}{B^u \ln\left(1 + \frac{p^t}{B^u N_0} \cdot G^u_{(1)}\right)}\right\}. \tag{8}$$

2) Downlink Transmission: When (k, r) code is adopted, the mobile device only needs to receive the results from r out of k edge nodes to recover the computation result of the original task [4]. With this favorable property of coded computing, edge node selection can be performed to achieve a diversity gain in downlink transmission. To this end, the r edge nodes with the highest channel power gains will be chosen for downlink transmission. In this work, time division multiple access is assumed, and hence the downlink transmission time is given by

$$T^d_{k,r} = \sum_{l=k-r+1}^{k} \min\left\{\frac{t^d_0}{r}, \frac{\zeta s/r}{B^d \cdot \ln\left(1 + \text{snr}^{\text{d}} \cdot G^d_{(l)}\right)}\right\}. \tag{9}$$

In (9), ζ is the ratio between the size of the computation result and that of the task; B^d is the bandwidth of the downlink; $\text{snr}^{\text{d}} \triangleq \frac{p^e}{B^d N_0}$ is the normalized SNR of the downlink with p^e the transmit power of the edge nodes; the order statistics [27] $G^d_{(l)}$ represents the lth smallest one among the k downlink channel power gains $\{G^d_i\}_{i=1}^k$.[6] As assumed at the beginning of this section, the size of a (k, r) encoded subtask is $1/r$ of that of the original task and so is the corresponding computation result. Hence, there is a factor of $1/r$ in (9). Similar to the uplink, it is assumed that each of the r downlink transmission times should not exceed $\frac{t^d_0}{r}$ so that the total downlink transmission time is within a predefined value t^d_0. Clearly, the corresponding downlink outage probability is determined by the distribution of the weakest channel power gain $G^d_{(k-r+1)}$ among the r chosen edge nodes and is given by

$$\epsilon^d_{\text{out}} = Pr\left\{\frac{t^d_0}{r} \le \frac{s\zeta/r}{B^d \ln\left(1 + \text{snr}^d \cdot G^d_{(k-r+1)}\right)}\right\}. \tag{10}$$

C. Problem Formulation

In the considered coded edge computing scenario, the mobile device aims to minimize the average processing time $\bar{T}_{k,r}$ of a task (c.f. (2)). According to the aforementioned models, $\bar{T}_{k,r}$ depends on multiple factors, including the local computing power p^l, the transmit power p^t, the offloading ratio ρ, as well as the task encoding parameters (k, r). Mathematically, the delay-optimal coded offloading problem can be formulated as follows

$$\mathbf{P1}: \min_{p^t, p^l, \rho, k, r} \bar{T}_{k,r} \tag{11}$$

$$\text{s.t. } \rho\epsilon^{\text{out}}_{k,r} \le \epsilon_0, \tag{12}$$

$$0 \le p^t, \quad p^l \le p_{\max},\ p^l + p^t \le p_{\max}, \tag{13}$$

$$0 \le \rho \le 1, \tag{14}$$

$$r, \quad k \in \{1, \ldots, n\},\ r \le k. \tag{15}$$

Some explanations are in order. The constraint (12) represents that the average outage probability $\rho\epsilon^{\text{out}}_{k,r}$ of all tasks should not exceed a predefined value ϵ_0; here, $\epsilon^{\text{out}}_{k,r} = \epsilon^u_{\text{out}} + \epsilon^d_{\text{out}} - \epsilon^u_{\text{out}}\epsilon^d_{\text{out}}$ represents the outage probability of the offloaded tasks. As the mobile device can simultaneously activate its local processing module and its transmit module, the total power consumption is subject to a constraint as in (13). The parameter n represents the maximum number of accessible edge nodes of the mobile device.

The above problem $\mathbf{P1}$ is a constrained non-linear mixed-integer programming, which is in general highly non-trivial to solve. Fortunately, by exploiting the inherent structure of $\mathbf{P1}$, it will be shown in Section IV that the optimal coding parameter r can be found via binary search. This allows us to develop a monotonic optimization based algorithm in Section V to find the optimal solution of $\mathbf{P1}$ efficiently.

[5]As the main focus is on the average task processing delay and the channel power gains over different timeslots are assumed to be independent and follow identical distributions, the corresponding subscripts of time in related quantities are omitted for ease of notation.

[6]For k i.i.d. random variables $U_1, \ldots, U_k$, their order statistics [27] will be written as $U_{(1)} \le U_{(2)} \le \cdots \le U_{(k)}$. Besides, when the uplink and the downlink channels are reciprocal, $G^d_i = G^u_i$ (for $i = 1, \ldots, n$) and so will be the corresponding order statistics.

IV. Optimal Coding Parameter Selection

In this section, by analyzing the inherent V-structure of the average processing delay w.r.t. the coding parameter r, efficient algorithms will be developed to find the optimal coding parameters.

Definition 1: A function $f(z)$ is said to have a *V-structure*, if there exists a z^* such that $f(z)$ is monotonic decreasing (increasing) when $z \le z^*$ ($z > z^*$).

Clearly, for any V-structured function f, $z^* = \arg\min f(z)$ and can be found via binary search. With this observation, it will be shown in subsection IV-A that, for any fixed k and power allocation p^l and p^t, the average processing delay of the offloaded tasks is a V-structure function of the coding parameter r when the outage constraint is not present. Hence, the outage constraint-free optimal $\mathring{r}$ can be found efficiently. Then, in subsection IV-B, it will be shown that $\mathring{r}$ is an upper bound of the optimal coding parameter r^* of the original constrained problem $\mathbf{P1}$. In addition, when $r \le \mathring{r}$, it will be shown that the objective function $\bar{T}_{k,r}$ of $\mathbf{P1}$ also admits the favorable V-structure w.r.t. r. This in turn allows an efficient binary search for the optimal coding parameter r^*.

Before presenting the detailed analysis, some useful facts are presented below to streamline the subsequent analysis. All the proofs will be presented in the appendices and the straightforward ones are omitted in the interest of space.

Fact 1: Suppose a function $f(u)$ admits $f(u) \ge 0$ ($f(u) \le 0$) on the interval $[a, c]$ ($[c, b]$) and $\int_a^c f(u)du + \int_c^b f(u)du = 0$. For any positive and non-increasing $g(u)$, $\int_a^b g(u)f(u)du \ge 0$.

Fact 2: For k i.i.d. random variables $U_1, \ldots, U_k$ with probability density function (p.d.f.) $f(u)$ and cumulative density function (c.d.f.) $F(u)$, the probability density function of the lth order statistics $U_{(l)}$ is given by

$$f_{U_{(l)}}(u) = kf(u)\binom{k-1}{l-1}F^{l-1}(u)(1-F(u))^{k-l}. \quad (16)$$

Fact 3: For any fixed positive integers k and $r \le k$, the equation $f(z) = 0$ only has one root z^* on $(0, 1)$, where

$$f(z) \triangleq \sum_{l=k-r-1}^{k} \frac{q(z;k,l)}{r+2} - \sum_{l=k-r}^{k} \frac{q(z;k,l)}{r+1} - \frac{r}{r+2}\left[\sum_{l=k-r}^{k} \frac{q(z;k,l)}{r+1} - \sum_{l=k-r+1}^{k} \frac{q(z;k,l)}{r}\right], \quad (17)$$

and

$$q(z;k,l) \triangleq \binom{k-1}{l-1} \cdot z^{l-1}(1-z)^{k-l}. \quad (18)$$

In addition, $f(z) \ge 0$ ($f(z) \le 0$) when $u \in (0, z^*]$ ($u \in [z^*, 1)$).

A. Coding Parameter Selection Without Outage Constraint

In this subsection, the coding parameter selection problem will be studied for the outage constraint-free case. Specifically, it will be shown that, under fixed k, the average processing time of the offloaded tasks admits a V-structure w.r.t. the coding parameter r.

1) Edge Computing Time: It follows from (5) that, for any arbitrary distribution of V_i's, the average time of waiting k edge nodes finishing their task computation is given by

$$\mathbb{E}\{T^e_{k,r}\} = \frac{s\beta}{rf^e_{\text{cpu}}} + \frac{1}{\lambda}\mathbb{E}\left\{V_{(k)}\right\}, \quad (19)$$

where the order statistics $V_{(k)}$ represents the largest one among all V_i's. The change rate of the average edge computing time w.r.t. r admits

$$\mathbb{E}\{T^e_{k,r}\} - \mathbb{E}\{T^e_{k,r+1}\} = \frac{s\beta}{f^e_{\text{cpu}}} \cdot \left(\frac{1}{r} - \frac{1}{r+1}\right). \quad (20)$$

This indicates that $\mathbb{E}\{T^e_{k,r}\}$ is a monotonically decreasing convex function of r.

2) Transmission Time: To analyze the transmission time of the uplink and the downlink in the (k, r) coded computing scenario, the p.d.f. and c.d.f. of the uplink (downlink) wireless channel power gains will be denoted by f_u (f_d) and F_u (F_d), respectively. The average uplink transmission time is given by

$$\mathbb{E}\left\{T^u_{k,r}\right\} = \mathbb{E}\left\{\left[\frac{s}{B^u \ln\left(1 + \text{snr}^u \cdot G^u_{(1)}\right)}\right]_{t^u_0}\right\}, \quad (21)$$

where the expectation is w.r.t. f_u and F_u and $[x]_a$ is a shorthand notation for $\min\{x, a\}$. Similarly, the average downlink transmission time is given by

$$\mathbb{E}\left\{T^d_{k,r}\right\} = \sum_{l=k-r+1}^{k} \mathbb{E}\left\{\left[\frac{s\zeta/r}{B^d \ln\left(1 + \text{snr}^d \cdot G^d_{(l)}\right)}\right]_{\frac{t^d_0}{r}}\right\}. \quad (22)$$

Lemma 1: For any fixed k, the change rate of the average downlink transmission time w.r.t. r admits

$$\left(\mathbb{E}\left\{T^d_{k,r+2}\right\} - \mathbb{E}\left\{T^d_{k,r+1}\right\}\right) - \frac{r}{r+2} \cdot \left(\mathbb{E}\left\{T^d_{k,r+1}\right\} - \mathbb{E}\left\{T^d_{k,r}\right\}\right) \ge 0. \quad (23)$$

Proposition 1: For any fixed k and power allocation, there exists an optimal $\mathring{r}$, such that $\mathbb{E}\{T^o_{k,r}\}$ (c.f. (1)) is monotonically decreasing (increasing) in r when $1 \le r \le \mathring{r}$ ($\mathring{r} \le r \le k$).

Remark 1: The insight revealed by Proposition 1 is that, under arbitrary fading distribution, the average task processing delay in the outage constraint-free case admits the favorable V-structure (c.f. Definition 1) w.r.t. the coding parameter. Consequently, the optimal (constraint-free) coding parameter $\mathring{r}$ can be efficiently found via binary search, as shown in Algorithm 1. Note that the complexity of Algorithm 1 is $\mathcal{O}(\log k)$ while the complexity of the brute-force search is $\mathcal{O}(k)$.

B. Optimal Coding Parameter With Outage Constraint

In this subsection, optimal coding parameter selection will be investigated for the original constrained problem $\mathbf{P1}$. Unless stated otherwise, all the discussions in this subsection assume fixed k and power allocation.

To this end, the influence of the coding parameter r on the outage probability will be analyzed first.

Algorithm 1 Binary Search for $\mathring{r}$

Initialize $r_L = 1$ and $r_U = k$
while $r_L < r_U$ **do**
  Set $\mathring{r} = \lfloor \frac{r_L + r_U}{2} \rfloor$;
  Compute $\mathbb{E}\{T^o_{k,r}\}$ for $\mathring{r}-1 \le r \le \mathring{r}+1$ by (19), (21), and (22);
  if $\mathbb{E}\{T^o_{k,\mathring{r}}\} \le \min\{\mathbb{E}\{T^o_{k,\mathring{r}-1}\}, \mathbb{E}\{T^o_{k,\mathring{r}+1}\}\}$ **then**
    Return $\mathring{r}$ and terminate the algorithm;
  else
    if $\mathbb{E}\{T^o_{k,\mathring{r}-1}\} \le \mathbb{E}\{T^o_{k,\mathring{r}}\} \le \mathbb{E}\{T^o_{k,\mathring{r}+1}\}$ **then**
      Set $r_L = \mathring{r} - 1$;
    else
      Set $r_U = \mathring{r} + 1$;
    end if
  end if
end while

Lemma 2: The outage probability $\epsilon^{out}_{k,r}$ is increasing in r.

For each r, consider the following optimization problem

$$\mathbf{P2}[r]: \min_{\rho} \ \bar{T}_{k,r} \tag{24}$$

$$\text{s.t. } \rho\epsilon^{\text{out}}_{k,r} \le \epsilon_0, \ 0 \le \rho \le 1. \tag{25}$$

Denote the optimal solution of $\mathbf{P2}[r]$ by $\rho^*(r)$ and the result below follows readily.

Lemma 3: For each r, the optimal offloading ratio is given by

$$\rho^*(r) = \min\left\{\frac{T^l}{T^l + \mathbb{E}\left\{T^o_{k,r}\right\}}, \frac{\epsilon_0}{\epsilon^{out}_{k,r}}\right\}, \tag{26}$$

and the corresponding optimal objective function of $\mathbf{P2}$ is given by[7]

$$\bar{T}_{k,r} = \max\left\{(1-\rho^*(r))\cdot T^l, \rho^*(r)\cdot \mathbb{E}\left\{T^o_{k,r}\right\}\right\}. \tag{27}$$

Note that the optimal r^* of the original problem $\mathbf{P1}$ under a given k and power allocation is given by $r^* = \arg\min_r \bar{T}_{k,r}$.

Proposition 2: (i) The optimal coding parameter $r^* \triangleq \arg\min_r \bar{T}_{k,r}$ admits $r^* \le \mathring{r}$,[8] where $\mathring{r}$ can be found by Proposition 1. (ii) $\bar{T}_{k,r}$ is monotonically decreasing (increasing) in r when $1 \le r \le r^*$ ($r^* \le r \le \mathring{r}$).

Remark 2: The insight revealed by Proposition 2 is that, under arbitrary fading, the task processing delay $\bar{T}_{k,r}$ admits the favorable V-structure w.r.t. the coding parameter r. This allows the optimal coding parameters r^* to be efficiently found via binary search as shown in Algorithm 2. Note that the complexity of Algorithm 2 is $\mathcal{O}(n\log(n))$ while that of the brute-force search is $\mathcal{O}(n^2)$.

V. Delay-Optimal Coded Offloading

In this section, the proposed delay-optimal coded offloading algorithm will be presented.

[7]For ease of presentation, a slight abuse of notation is taken here. Particularly, $\bar{T}_{k,r}$ defined in (27) can be viewed as the $\bar{T}_{k,r}$ defined in (2) with optimal offloading ratio.

[8]When more than one optimal r's exist, r^* is defined as the smallest one.

Algorithm 2 Low-Complexity Search for (k^*, r^*)

for $k = 1, \ldots, n$ **do**
  Call Algorithm 1 to find $\mathring{r}$ given k;
  Initialize $r_L = 1$ and $r_U = \mathring{r}$
  while $r_L < r_U$ **do**
    Set $r_k = \lfloor \frac{r_L + r_U}{2} \rfloor$;
    For $r_k - 1 \le r \le r_k + 1$, compute $\bar{T}_{k,r_k}$ by (27);
    if $\bar{T}_{k,r_k} \le \min\{\bar{T}_{k,r_k-1}, \bar{T}_{k,r_k+1}\}$ **then**
      Record r_k and terminate the current loop for r;
    else
      if $\bar{T}_{k,r_k-1} \le \bar{T}_{k,r_k} \le \bar{T}_{k,r_k}$ **then**
        Set $r_L = r_k - 1$;
      else
        Set $r_U = r_k + 1$;
      end if
    end if
  end while
  Set $T_k = \bar{T}_{k,r_k}$;
end for
Set $k^* = \arg\min T_k$ and $r^* = r_{k^*}$.

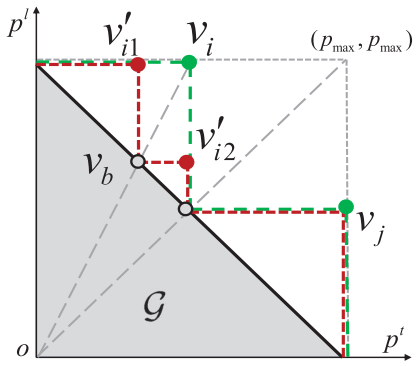


Fig. 2. Illustration of the Polyblock outer approximation method.

The overall structure of the proposed algorithm is as follows. First, the original delay-optimal coded offloading problem $\mathbf{P1}$ will be converted into a (two-dimensional) monotonic optimization w.r.t. the power allocation p^t and p^l. Within each iteration of the monotonic optimization, the corresponding optimal coding parameters and offloading ratio under fixed power allocation (p^t, p^l) can be found according to the analysis in Section IV.

A. Preliminary on Monotonic Optimization

To facilitate understanding of the proposed algorithm, some preliminaries of monotonic optimization are reviewed.

Definition 2: A set $\mathcal{G}$ is *normal* if for any $\mathbf{x} \in \mathcal{G}$, all other points $\mathbf{x}'$ such that $\mathbf{0} \preceq \mathbf{x}' \preceq \mathbf{x}$ are in $\mathcal{G}$.

Definition 3: When $f(\mathbf{x})$ is monotonic decreasing in $\mathbf{x}$ and $\mathcal{G}$ is a compact normal set with non-empty interior, the

following problem is a *monotonic optimization* [28]

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{28}$$

$$\text{s.t. } \mathbf{x} \in \mathcal{G}. \tag{29}$$

The above monotonic optimization problem can be solved optimally using the Polyblock outer approximation method [28] as illustrated in Fig. 2. In particular, as the objective function $f(\mathbf{x})$ is monotonically decreasing, a set of vertices $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots\}$ are maintained to keep track of a lower bound $f_L = \min\{f(\mathbf{v}_1), f(\mathbf{v}_2), \ldots\}$ of $f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is the optimal solution of this monotonic optimization problem. At each iteration, a vertex $\mathbf{v}_i = \arg\min\{f(\mathbf{v}_1), f(\mathbf{v}_2), \ldots\}$ is chosen, and the corresponding boundary point $\mathbf{v}_b \in \partial\mathcal{G}$ on the line segment $\mathbf{ov_i}$ is found; here, $\partial\mathcal{G}$ denotes the boundary of $\mathcal{G}$. Clearly, $f_U = f(\mathbf{v}_b)$ is an upper bound of $f(\mathbf{x}^*)$. Then, project $\mathbf{v}_b$ onto the boundary of the current polyblock (the green one) as in Fig. 2 and update the vertex set $\mathcal{V}$ by removing $\mathbf{v}_i$ and adding the two projection vertices $\mathbf{v}'_{i1}$ and $\mathbf{v}'_{i2}$. This will generate the new polyblock (the red one). The above procedure will repeat until the gap $f_U - f_L$ becomes sufficiently small. The convergence of this algorithm readily follows from the fact that the gap $f_U - f_L$ monotonically decreases after each iteration. More details and variations about the Polyblock outer approximation method can be found in [28].

B. Offloading Parameter Optimization

Through the following discussions, it will be shown that the delay-optimal coded offloading problem $\mathbf{P1}$ is a monotonic optimization.

For ease of presentation, for any power allocation (p^t, p^l), denote the average task processing time under optimal coding parameters and offloading ratio selection by $T(p^t, p^l)$. More specifically, let $(\hat{k}, \hat{r})$ be the optimal coding parameters found by Algorithm 2 under power allocation (p^t, p^l) and $\hat{\rho}$ be the corresponding offloading ratio given by (26). Then, $T(p^t, p^l)$ is given by

$$T(p^t, p^l) \triangleq \min_{k,r,\rho} \bar{T}_{k,r} = \max\left\{(1-\hat{\rho})T^l, \hat{\rho}\mathbb{E}\left\{T^o_{\hat{k},\hat{r}}\right\}\right\}. \tag{30}$$

With this notation, $\mathbf{P1}$ can be rewritten as a monotonic optimization as follows:

$$\mathbf{P2}: \min_{p^t, p^l} T(p^t, p^l) \tag{31}$$

$$\text{s.t. } 0 \le p^t, p^l \le p_{\max},\ p^l + p^t \le p_{\max}. \tag{32}$$

Fact 4: $\mathbf{P2}$ is a monotonic optimization problem w.r.t. the power allocation (p^t, p^l).

Consequently, $\mathbf{P2}$ can be solved optimally by the aforementioned Polyblock outer approximation method. When performing the Polyblock outer approximation (c.f. Fig. 2), for any vertex (p^t, p^l), it follows from the geometric properties that the coordinate of the corresponding boundary point v_b is given by $\left(\frac{p^t p_{\max}}{p^t+p^l}, \frac{p^l p_{\max}}{p^t+p^l}\right)$. Similarly, the coordinates of the two new projection vertices are given by $\left(p^t, \frac{p^l p_{\max}}{p^t+p^l}\right)$ and $\left(\frac{p^t p_{\max}}{p^t+p^l}, p^l\right)$, respectively.

With the above results, the delay-optimal coded offloading problem $\mathbf{P1}$ can be solved optimally by Algorithm 3.

Algorithm 3 Finding Optimal Offloading Parameters

1: Initialization: $f_U = \infty$ and $\mathcal{V} = \{(p_{\max}, p_{\max})\}$
2: Find $T(p_{\max}, p_{\max})$ by (30);
3: Set $f_L = \mathbb{E}\{T_{k,r}(p_{\max}, p_{\max})\}$ and $V(p_{\max}, p_{\max}) = f_L$;
4: **while** $f_U - f_L > \epsilon_{th}$ **do**
5: Find vertex $v = \arg\min V \in \mathcal{V}$ and write $v = (p^t, p^l)$;
6: Find boundary vertex $v_b = \left(\frac{p^t p_{\max}}{p^t+p^l}, \frac{p^l p_{\max}}{p^t+p^l}\right)$;
7: Find $T(v_b)$ by (30);
8: **if** $f_U \le \mathbb{E}\{T_{k,r}(v_b)\}$ **then**
9: Set $f_U = \mathbb{E}\{T_{k,r}(v_b)\}$;
10: Set v_b as the optimal power allocation;
11: **end if**
12: Remove v from set $\mathcal{V}$;
13: Add two projection vertices $\left(p^t, \frac{p^l p_{\max}}{p^t+p^l}\right)$ and $\left(\frac{p^t p_{\max}}{p^t+p^l}, p^l\right)$ to $\mathcal{V}$;
14: **end while**

VI. Simulation Results

In this section, simulation results are presented to corroborate the effectiveness of the proposed coded offloading scheme.

In the simulations, the size of each computation task is set to 1×10^3 bits and each bit requires $\beta = 2000$ CPU cycles; the maximum local computing frequency of the mobile device is set to $f^l_{\max} = 1$ (GHz) and the coefficient κ is set to 1×10^{-28}; the processor frequency of the edge nodes is set to $f^e_{\text{cpu}} = 1$ (GHz); the ratio ζ between the size of the computation result and that of the computation task is set to 5. The uplink and downlink bandwidths are set to 2 (MHz); the power spectral density of the channel noise is set to $N_0 = 1 \times 10^{-17}$ (W/Hz); the scaling factor λ of the exponentially distributed straggling effect is set to 1×10^5; the total power constraint of the mobile device $p_{\max}$ is set to 1 (W) and the transmit power of the edge nodes is set to 0.5 (W); the uplink and the downlink outage transmission times t^u_0 and t^d_0 are set to 2×10^{-3} (sec); the outage threshold ϵ_0 is set to 0.01. The maximum number of available edge nodes is set to $n = 20$. To demonstrate the advantage of the proposed coded offloading scheme, four baseline schemes are considered. The first baseline is the conventional single edge node offloading scheme, in which the coding (k, r) parameter is $(1, 1)$. The second baseline is the conventional distributed edge computing scheme, which is equivalent to setting (k, r) to (n, n). In the third baseline, the optimal coding parameter is taken but no edge node selection is performed. In the fourth baseline, edge node selection is performed but

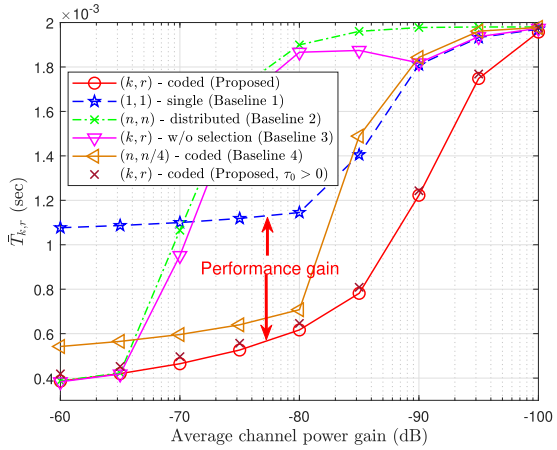


Fig. 3. Comparison of average processing time. (Rayleigh fading.)

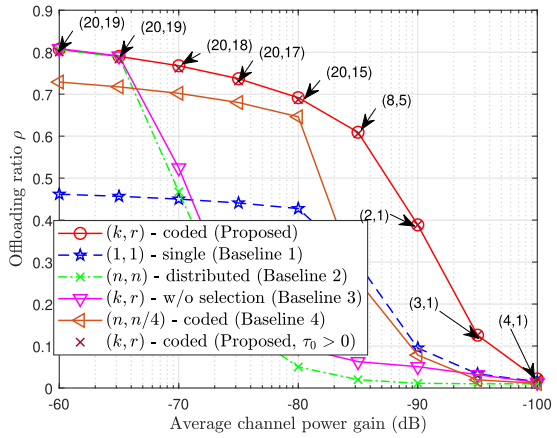


Fig. 4. Comparison of offloading. (Rayleigh fading.)

the employed coding parameter is not optimal.[9,10] In addition, the performance of the proposed scheme with a non-zero task decoding time $\tau_0 = 2 \times 10^{-5}$ (sec) is evaluated.[11]

The performances in Rayleigh fading [5] environments are presented in Figs. 3–5. Specifically, the average task processing time $\bar{T}_{k,r}$'s of the proposed scheme and the two baselines are compared in Fig. 3 under different average channel power gains. It can be seen that the proposed scheme can reduce the task processing time $\bar{T}_{k,r}$ substantially. For example, when the average channel power gain is -80 (dB), the average task processing time $\bar{T}_{k,r}$ of the two baselines are 1.8×10^{-3} (sec) and 1.18×10^{-3} (sec), respectively, while the proposed scheme reduces $\bar{T}_{k,r}$ to about 0.6×10^{-3} (sec). When the average channel power gain is -82 (dB), the average processing time of Baselines 3 and 4 are about

[9]When the statistics of the wireless environment (and other system parameters) remain unchanged, the proposed optimization algorithm only needs to be executed once at initialization. Hence, its running time is not included in the simulations.

[10]As in the literatures (e.g., [12], [14]), this work considers general (k,r) linear codes and assumes that the result of the original task can be recovered so long as r edge nodes return their feedbacks. Hence, the details of task encoding/decoding are not considered in the simulations.

[11]This value is obtained from our empirical experiments based on MATLAB implementation.

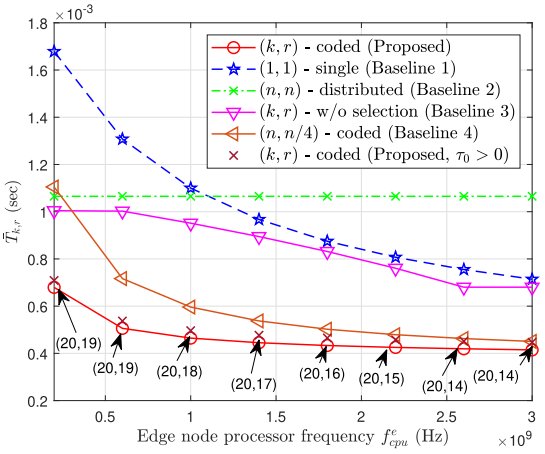


Fig. 5. Comparison of average processing time. (Rayleigh fading.)

1.9×10^{-3} and 1.5×10^{-3} (sec), respectively, while that of the proposed scheme is only about 0.8×10^{-3} (sec). Besides, it can be seen that when the average channel power gain is large (e.g., at -60 (dB)), the performance of the proposed scheme coincides with those of Baselines 2 and 3. The reason is that when the channel power gain is large, the transmission delay caused by fading becomes negligible as compared to the computation time at the edge nodes. On the other extreme, when the average channel power gain is low (e.g., at -100 (dB)), the performances of all these schemes become identical. The reason is that, as shown in Fig. 4, when the channel condition is poor, the amount of tasks offloaded by the mobile device vanishes and the average task processing time $\bar{T}_{k,r}$ will be mainly determined by its local computation time T^l. Also, it can be observed that the proposed scheme with task decoding time τ_0 included (c.f. the dark red $\times$'s) still outperforms the baselines. The optimal coding parameters are also presented in Fig. 4. For example, when the average channel power gain is -70 (dB), the optimal coding parameter (k,r) is $(20,18)$. Besides, it can be seen that more edge nodes will be accessed (i.e., larger values of k) by the proposed scheme when the channel condition is better.

The performances of these schemes are compared under different edge node processor frequencies in Fig. 5, where the average channel power gain is set to -70 (dB). It can be seen that under different f^e_{cpu}'s, the proposed scheme substantially outperforms the four baseline schemes and the average task processing time $\bar{T}_{k,r}$ monotonically decreases as f^e_{cpu} increases. The corresponding optimal coding parameter (k,r)'s are also presented in Fig. 5. It can be seen that the value of r monotonically decreases as f^e_{cpu} increases. The reason is that, when the computing capability of the edge nodes increases, the proposed scheme will put more emphasis on improving the transmission diversity gain by enlarging the difference between k and r.

The performance comparisons for other fading scenarios are presented in Figs. 6–8. In particular, when the channel power gains are distributed according to Rician fading $\mathcal{R}(K_r, \Omega_r)$, the corresponding average task processing times are compared in Fig. 6 with different Rician factors K_r's. It can be seen

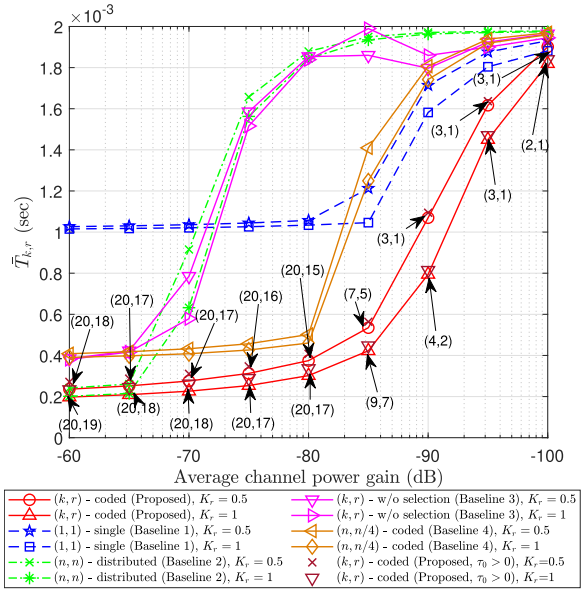


Fig. 6. Comparison of average processing time.(Rician fading.)

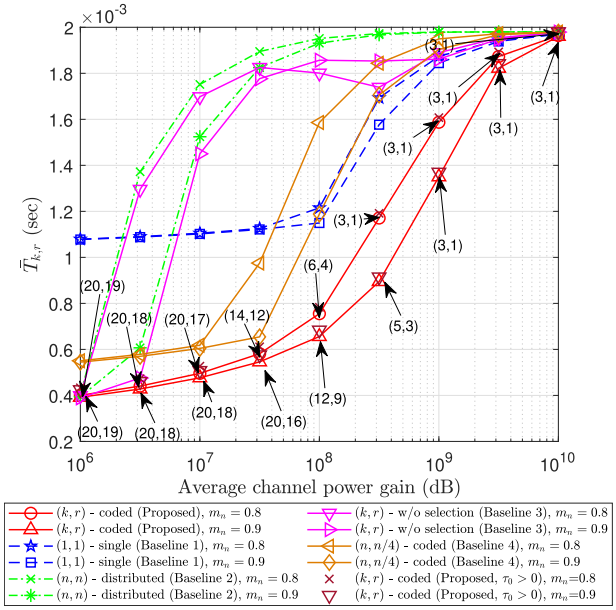


Fig. 7. Comparison of average processing time. (Nakagami fading.)

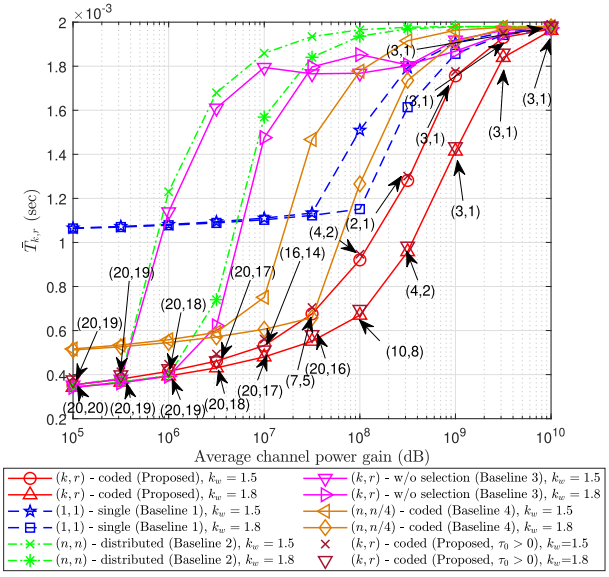


Fig. 8. Comparison of average processing time. (Weibull fading.)

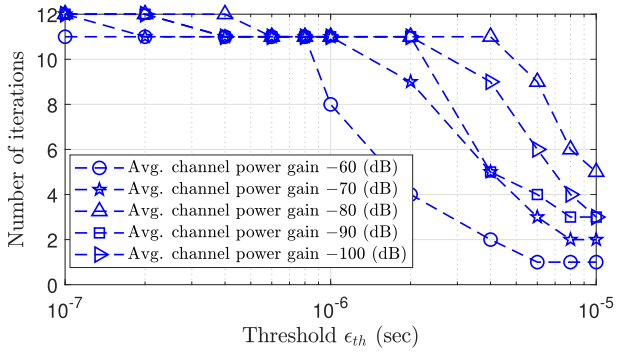


Fig. 9. Convergence speed of the proposed algorithm.

that the performance gain brought by the proposed scheme is significant. For example, when the average channel power gain is -82 (dB), the proposed scheme can reduce $\bar{T}_{k,r}$ to about 0.5×10^{-3} (sec), leading to about 60% performance improvement as compared to the conventional single edge node offloading. Similar observations can be made for the Nakagami fading $\mathcal{N}(m_n, \Omega_n)$ [5] and the Weibull fading $\mathcal{W}(k_w, \lambda_w)$ [41] from Fig. 7 and Fig. 8, respectively. These results validate that the proposed scheme remains effective in different fading environments with general SNRs.

The convergence rate of the proposed Polyblock based optimization algorithm is evaluated. Particularly, it can be seen from Fig. 9 that the proposed algorithm converges fairly quickly. For example, when the threshold ϵ_{th} in Algorithm 3 is 1×10^{-6} (sec), which is pretty small as compared to $\bar{T}_{k,r}$, the proposed algorithm converges in less than 12 iterations under different average channel power gains.

VII. Conclusion and Future Works

In this work, a delay-optimal coded offloading scheme is proposed, in which the offloaded computation tasks are encoded by (k, r) codes and transmission diversity gains are obtained by performing edge node selection to mitigate fading. Besides, through in-depth analysis based on order statistics, it is found that the average processing delay of the offloaded tasks admits a favorable V-structure w.r.t. the coding parameter r under arbitrary fading distributions. Based on this key theoretic result, an efficient monotonic optimization based algorithm is developed to find the optimal coding, computation, and communication parameters for the proposed coded offloading scheme. Simulation results show that the proposed scheme can substantially outperform the conventional single edge node offloading and distributed edge offloading schemes and greatly reduce the average task processing delay.

Worthwhile future works include studying the energy-optimal counterpart of the proposed scheme, incorporating

task encoding in the uplink transmission, allowing parallel execution of multiple tasks on edge nodes, and integrating the proposed scheme with other collaborative transmission schemes (e.g., over-the-air computation and interference alignment).

APPENDIX A
PROOF OF FACT 2

Proof: This is a known result in order statistics [27] and the proof is included here for the completeness.

$$\begin{aligned} f_{U_{(l)}}(u) &= \lim_{\epsilon\to 0}\mathbb{P}\{U_{(l)}\in[u,u+\epsilon]\} \\ &= \lim_{\epsilon\to 0}\sum_{i=1}^{k}\mathbb{P}\{U_i\in[u,u+\epsilon], E_{l-1,\{1,\ldots,k\}\setminus\{i\}}(u)\} \\ &= \lim_{\epsilon\to 0} k\cdot\binom{k-1}{l-1} \\ &\quad\cdot\mathbb{P}\{U_1\in[u,u+\epsilon]\}\cdot\mathbb{P}^{l-1}\{U_2<u\} \\ &\quad\cdot\mathbb{P}^{k-1}\{U_2>u\} \\ &= kf(u)\binom{k-1}{l-1}F^{l-1}(u)(1-F(u))^{k-l}, \end{aligned} \tag{33}$$

where the assumed i.i.d. property is invoked in the first and the second equalities; $E_{l-1,\{1,\ldots,k\}\setminus\{i\}}(u)$ represents the event that exactly $l-1$ random variables among $\{U_j\}_{j=1,j\neq i}^{k}$ are less than u. ∎

APPENDIX B
PROOF OF FACT 3

Proof: It can be verified that

$$\begin{aligned} f(z) &= z^{k-r-2}(1-z)^r\cdot(a-bz) \\ &\quad+\underbrace{\left(\frac{1}{r+2}-\frac{1}{r+1}-\frac{r\cdot\left(\frac{1}{r+1}-\frac{1}{r}\right)}{r+2}\right)}_{=0} \\ &\quad\cdot\sum_{l=k-r+1}^{k}q(z;k,l), \end{aligned} \tag{34}$$

where the coefficients a and b are given by

$$a=\frac{1}{r+2}\cdot\binom{k-1}{k-r-2}>0, \tag{35}$$

and

$$b=\frac{1}{r+2}\cdot\left(\binom{k-1}{k-r-2}+\binom{k-1}{k-r-1}\right)>0. \tag{36}$$

With the above observation, the results of this fact readily follows. ∎

APPENDIX C
PROOF OF LEMMA 1

Proof: First notice that

$$\begin{aligned} &\mathbb{E}\left\{T_{k,r}^d\right\} \\ &= \sum_{l=k-r+1}^{k}\mathbb{E}\left\{\frac{s\zeta/r}{B^d\ln(1+\mathrm{snr}^d\cdot G_{(l)}^d)}\right\} \\ &= \frac{1}{r}\cdot\frac{s\zeta}{B^d}\sum_{l=k-r+1}^{k}\int_0^\infty\frac{1}{\ln(1+\mathrm{snr}^d\cdot u)}\cdot k\cdot\binom{k-1}{l-1} \\ &\quad\cdot f_G(u)F_G^{l-1}(u)(1-F_G(u))^{k-l}du \\ &= \frac{s\zeta}{B^d}\int_0^1\frac{1}{\ln(1+\mathrm{snr}^d\cdot F_G^{-1}(z))}\cdot\phi_{k,r}(z)dz, \end{aligned} \tag{37}$$

where the change of variable $F_G(u)=z$ is applied in the second equality and

$$\phi_{k,r}(u)\triangleq\frac{k}{r}\sum_{l=k-r+1}^{k}\binom{k-1}{l-1}\cdot z^{l-1}(1-z)^{k-l}. \tag{38}$$

With the above observation, it follows that

$$\begin{aligned} &\text{LHS of (23)} \\ &= \int_0^1\frac{s\zeta}{B^d\ln(1+\mathrm{snr}^d\cdot F_G^{-1}(z))}\cdot\Big(\phi_{k,r+2}(z) \\ &\quad-\phi_{k,r+1}(z)-\frac{r}{r+2}\left(\phi_{k,r+1}(z)-\phi_{k,r}(z)\right)\Big)du. \end{aligned} \tag{39}$$

Note that

$$\begin{aligned} \int_0^1\phi_{k,r}(z)dz &= \frac{1}{r}\sum_{l=k-r+1}^{k}\int_0^\infty k\cdot\binom{k-1}{l-1}\cdot f_G(u) \\ &\quad\cdot F_G^{l-1}(u)(1-F_G(u))^{k-l}du \\ &= \frac{1}{r}\sum_{l=k-r+1}^{k}\int_0^\infty f_{U_{(l)}}(u)du=1, \end{aligned} \tag{40}$$

where $f_{U_{(l)}}$ is the p.d.f. defined in Fact 2. Then, it follows that

$$\begin{aligned} &\int_0^1\phi_{k,r+2}(z)-\phi_{k,r+1}(z) \\ &\qquad-\frac{r}{r+2}\left(\phi_{k,r+1}(z)-\phi_{k,r}(z)\right)dz=0. \end{aligned} \tag{41}$$

It follows from Fact 3 that the function $\phi_{k,r+2}(z)-\phi_{k,r+1}(z)-\frac{r}{r+2}\left(\phi_{k,r+1}(z)-\phi_{k,r}(z)\right)$ is positive and negative on the intervals $(0,z^*)$ and $[z^*,1)$, respectively. Besides, it can be verified that $\frac{1}{\ln(1+\mathrm{snr}^d\cdot F_G^{-1}(z))}$ is positive and decreasing on $(0,1)$. Combining these two observations and (41) and then invoking Fact 1, it is not difficult to see that the above integral is non-negative. This completes the proof. ∎

APPENDIX D
PROOF OF PROPOSITION 1

Proof: If $\mathbb{E}\{T_{k,r}^o\}$ is monotonically decreasing in r, then $r^*=k$. Otherwise, it is sufficient to shown that if $\mathbb{E}\{T_{k,r}^o\}<\mathbb{E}\{T_{k,r+1}^o\}$ for some $r<k$, then $\mathbb{E}\{T_{k,r+1}^o\}<\mathbb{E}\{T_{k,r+2}^o\}$. To this end, notice that $\mathbb{E}\{T_{k,r}^o\}<\mathbb{E}\{T_{k,r+1}^o\}$ implies

$$\mathbb{E}\{T_{k,r}^e\}-\mathbb{E}\{T_{k,r+1}^e\}<\mathbb{E}\{T_{k,r+1}^d\}-\mathbb{E}\{T_{k,r}^d\}. \tag{42}$$

Consequently, it has

$$\begin{aligned} &\mathbb{E}\{T_{k,r+1}^o\}-\mathbb{E}\{T_{k,r+2}^o\} \\ &= \mathbb{E}\{T_{k,r+1}^e\}-\mathbb{E}\{T_{k,r+2}^e\}+\left(\mathbb{E}\{T_{k,r+1}^d\}-\mathbb{E}\{T_{k,r+2}^d\}\right) \\ &= \left(\mathbb{E}\{T_{k,r}^e\}-\mathbb{E}\{T_{k,r+1}^e\}\right)\cdot\frac{r}{r+2} \\ &\quad+\left(\mathbb{E}\{T_{k,r+1}^d\}-\mathbb{E}\{T_{k,r+2}^d\}\right) \end{aligned}$$

$$\leq \left(\mathbb{E}\{T^e_{k,r}\} - \mathbb{E}\{T^e_{k,r+1}\}\right) \cdot \frac{r}{r+2} + \left(\mathbb{E}\{T^d_{k,r}\} - \mathbb{E}\{T^d_{k,r+1}\}\right) \cdot \frac{r}{r+2} \leq 0, \quad (43)$$

where the second equality follows from (20) and the last two inequalities follows from Lemma 1 and (42). ∎

Appendix E
Proof of Lemma 2

Proof: As ϵ^u_{out} does not depend on r, it is sufficient to show ϵ^d_{out} increases in r. To this end, it follows from (10) that ϵ^d_{out} is increased by $\Delta\epsilon$ when r is increased to $r+1$, where $\Delta\epsilon$ is given by

$$\begin{aligned}\Delta\epsilon &= Pr\left\{\frac{t^d_0}{r+1} \leq \frac{s\zeta/(r+1)}{B^d \ln\left(1+\mathrm{snr}^d \cdot G^d_{(k-r)}\right)}\right\} \\ &\quad - Pr\left\{\frac{t^d_0}{r} \leq \frac{s\zeta/r}{B^d \ln\left(1+\mathrm{snr}^d \cdot G^d_{(k-r+1)}\right)}\right\} \\ &= \int_0^{F_d(u_0)} H_{k,r}(z)dz,\end{aligned} \quad (44)$$

where $u_0 \triangleq \frac{1}{\mathrm{snr}^d}\left(\exp\left(\frac{s\zeta}{t^d_0 B^d}\right)-1\right)$ and $H(z;k,r)$ is defined as

$$H_{k,r}(z) \triangleq k\binom{k-1}{k-r-1} z^{k-r-1}(1-z)^{r-1}\left[1-\frac{k}{k-r}z\right]. \quad (45)$$

In (44), the last equality follows from Fact 2 and change of variable $F_d(u) = z$.

Since $H_{k,r}(z)$ is actually the difference between the p.d.f.'s of $G^d_{(k-r)}$ and $G^d_{(k-r+1)}$ (after the change of variable), it is clear that $\int_0^1 H_{k,r}(z)dz = 0$. Besides, $H_{k,r}(z)$ is positive on $[0, \frac{k-r}{k})$ and negative on $(\frac{k-r}{k}, 1]$, respectively. Hence, when $F_d(u_0) \leq \frac{k-r}{k}$, it follows that $\int_0^{F_d(u_0)} H_{k,r}(z)dz > \int_0^{F_d(u_0)} 0\ dz = 0$. When $F_d(u_0) > \frac{k-r}{k}$, it follows that $\int_0^{F_d(u_0)} H_{k,r}(z)dz = 0 - \int_{F_d(u_0)}^1 H_{k,r}(z)dz > 0 - \int_{F_d(u_0)}^1 0\ dz = 0$. This completes the proof. ∎

Appendix F
Proof of Proposition 2

Proof: Part (i) will be proved by contradiction. Specifically, it will be shown that if $r^* > \mathring{r}$, one can always find an r such that $r^* > r \geq \mathring{r}$ and $\bar{T}_{k,r} \leq \bar{T}_{k,r^*}$. To this end, let $r = r^* - 1$ and it follows that

$$\begin{aligned}&\bar{T}_{k,r^*-1} \\ &= \max\left\{(1-\rho^*(r^*-1))\cdot T^l, \rho^*(r^*-1)\cdot \mathbb{E}\left\{T^o_{k,r^*-1}\right\}\right\} \\ &\leq \max\left\{(1-\rho^*(r^*))\cdot T^l, \rho^*(r^*)\cdot \mathbb{E}\left\{T^o_{k,r^*-1}\right\}\right\} \\ &\leq \max\left\{(1-\rho^*(r^*))\cdot T^l, \rho^*(r^*)\cdot \mathbb{E}\left\{T^o_{k,r^*}\right\}\right\} = \bar{T}_{k,r^*},\end{aligned} \quad (46)$$

where the first and the last equalities follow from the definitions of $\bar{T}_{k,r}$ in (27); the first inequality follows from the optimality of $\rho^*(r^*-1)$; the second inequality follows from the fact that $\mathbb{E}\left\{T^o_{k,r}\right\}$ is increasing in r when $r \geq \mathring{r}$ (c.f. Proposition 1).

To see part (ii), it is equivalent to show that $\bar{T}_{k,r-1} \geq \bar{T}_{k,r}$ always implies $\bar{T}_{k,r-2} \geq \bar{T}_{k,r-1}$ when $r \leq \mathring{r}$. For the ease of notation, define $\mathring{\rho}_r = \frac{T^l}{T^l + \mathbb{E}\left\{T^o_{k,r}\right\}}$. Firstly, it will be shown by contradiction that $\mathring{\rho}_{r-1} \leq \frac{\epsilon_0}{\epsilon^{out}_{k,r-1}}$. Suppose this is not true, then $\mathring{\rho}_{r-1} > \frac{\epsilon_0}{\epsilon^{out}_{k,r-1}}$. In this case, it can be verified that $\rho^*(r-1) = \frac{\epsilon_0}{\epsilon^{out}_{k,r-1}}$ and $(1-\rho^*(r-1))\cdot T^l \geq \rho^*(r-1)\cdot \mathbb{E}\left\{T^o_{kr-1}\right\}$. Hence,

$$\bar{T}_{k,r-1} = \left(1-\frac{\epsilon_0}{\epsilon^{out}_{k,r-1}}\right)\cdot T^l \leq \left(1-\frac{\epsilon_0}{\epsilon^{out}_{k,r}}\right)\cdot T^l \leq \bar{T}_{k,r}, \quad (47)$$

where the first inequality follows from the increasing property of $\epsilon^{out}_{k,i}$ and the second inequality follows from (27). This leads to a contradiction.

Since it has been shown above that $\mathring{\rho}_{r-1} \leq \frac{\epsilon_0}{\epsilon^{out}_{k,r-1}}$. It follows that

$$\mathring{\rho}_i \leq \mathring{\rho}_{r-1} \leq \frac{\epsilon_0}{\epsilon^{out}_{k,r-1}} \leq \frac{\epsilon_0}{\epsilon^{out}_{k,i}}, \quad \forall i \leq r-1, \quad (48)$$

where the first inequality follows from the fact that $\mathbb{E}\left\{T^o_{k,i}\right\}$ is decreasing when $i < \mathring{r}$ (c.f. Proposition 1) and the fact that $\mathring{\rho}_i$ is decreasing w.r.t. $\mathbb{E}\left\{T^o_{k,i}\right\}$; the last inequality follows from Lemma 2. Hence, by (26), $\rho^*_i = \mathring{\rho}_i$ for all $i \leq r-1$, which implies that

$$\bar{T}_{k,i} = \frac{\mathbb{E}\left\{T^o_{k,i}\right\}}{T^l + \mathbb{E}\left\{T^o_{k,i}\right\}}\cdot T^l, \quad \forall i \leq r-1. \quad (49)$$

Since $\mathbb{E}\left\{T^o_{k,i}\right\}$ is decreasing when $i \leq \mathring{r}$, $\bar{T}_{k,i}$ is decreasing for all $i < r-1$. Hence, $\bar{T}_{k,r-2} \geq \bar{T}_{k,r-1}$. ∎

Appendix G
Proof of Fact 4

Proof: In particular, it follows from (2), (3), and (7) that $T(p^t, p^l)$ is monotonically decreasing in the local computing power p^l. Besides, for any $p^t_1 \geq p^t_2$ and fixed p^l, denote the corresponding optimal coding parameters as k^*_1, r^*_1 and k^*_2, r^*_2, respectively, and optimal offloading ratios as, ρ^*_1 and ρ^*_2, respectively. It is clear that

$$\begin{aligned}T(p^t, p^l_1) &= \mathbb{E}\left\{T_{k,r}(p^t_1, p^l)|\rho^*_1, k^*_1, r^*_1\right\} \\ &\leq \mathbb{E}\left\{T_{k,r}(p^t_1, p^l)|\rho^*_2, k^*_2, r^*_2\right\} \\ &\leq \mathbb{E}\left\{T_{k,r}(p^t_2, p^l)|\rho^*_2, k^*_2, r^*_2\right\} = T(p^t_2, p^l),\end{aligned} \quad (50)$$

where the first inequality follows from the optimality of ρ^*_1, k^*_1, r^*_1 when the transmit power is p^t_1; the second inequality follows from that the uplink tranmission time (7) is non-increasing w.r.t. the transmit power p^t. Note that $(p^t_1, p^l, \rho^*_2, k^*_2, r^*_2)$ is a feasible configuration, since the feasible region of ρ is enlarged when the transmit power increases from p^t_2 to p^t_1. On the other hand, it is clear that the set of feasible power allocations $\mathcal{G} = \{(p^t, p^l)|0 \leq p^t, p^l \leq p_{\max}, p^t + p^l \leq p_{\max}\}$ is a normal set. Therefore, $\mathbf{P2}$ is a monotonic optimization. ∎

REFERENCES

[1] 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, 4th Quart., 2017.

[2] A. Jonathan, M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: Distributed edge cloud for data intensive computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3229–3242, Nov. 2017.

[3] K. Taik-Kim, C. Joe-Wong, and M. Chiang, "Coded edge computing," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, Jul. 2020, pp. 237–246.

[4] J. S. Ng, W. Y. B. Lim, N. C. Luong, and Z. Xiong, "A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1800–1837, 3rd Quart., 2021.

[5] A. Goldsmith, *Wireless Communications*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, Aug. 2005.

[6] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Communication-aware computing for edge processing," in *Proc. IEEE ISIT*, Aachen, Germany, Jun. 2017, pp. 2885–2889.

[7] J. Yue and M. Xiao, "Coding for distributed fog computing in internet of mobile things," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1337–1350, Apr. 2021.

[8] S. Zhao, "A node-selection-based sub-task assignment method for coded edge computing," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 797–801, May 2019.

[9] B. Wang, J. Xie, K. Lu, Y. Wan, and S. Fu, "Coding for heterogeneous UAV-based networked airborne computing," in *Proc. IEEE GLOBECOM Workshops*, Waikoloa, HI, USA, Dec. 2020, pp. 1–6.

[10] Y. Keshtkarjahromi, Y. Xing, and H. Seferoglu, "Dynamic heterogeneity-aware coded cooperative computation at the edge," in *Proc. IEEE ICNP*, Sep. 2018, pp. 23–33.

[11] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.

[12] K. Li, M. Tao, and Z. Chen, "Exploiting computation replication for mobile edge computing: A fundamental computation-communication tradeoff study," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4563–4578, Jul. 2020.

[13] L. Shi, K. Cai, and Z. Mei, "Linear network coded computation in mobile edge computing," in *Proc. IEEE GLOBECOM*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[14] J. Zhang and O. Simeone, "On model coding for distributed inference and transmission in mobile edge computing systems," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1065–1068, Jun. 2019.

[15] D. Kim, H. Park, and J. K. Choi, "Optimal load allocation for coded distributed computation in heterogeneous clusters," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 44–58, Jan. 2021.

[16] C. S. Yang, R. Pedarsani, and A. S. Avestimehr, "Coded computing in unknown environment via online learning," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020, pp. 185–190.

[17] B. Hasircioğlu, J. Gómez-Vilardebó, and D. Gündüz, "Bivariate polynomial coding for straggler exploitation with heterogeneous workers," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020, pp. 251–256.

[18] N. Raviv, Q. Yu, J. Bruck, and S. Avestimehr, "Download and access trade-offs in Lagrange coded computing," in *Proc. IEEE ISIT*, Paris, France, Jul. 2019, pp. 1787–1791.

[19] J. Kosaian, K. V. Rashmi, and S. Venkataraman, "Learning-based coded computation," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 227–236, May 2020.

[20] S. Prakash, S. Dhakal, M. Akdeniz, Y. Yona, and N. Himayat, "Coded computing for low-latency federated learning over wireless edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 233–250, Jan. 2021.

[21] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE ISIT*, Jul. 2016, pp. 1451–1455.

[22] X. He, R. Jin, and H. Dai, "Physical-layer assisted secure offloading in mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4054–4066, Jun. 2020.

[23] A. Reisizadeh and R. Pedarsani, "Latency analysis of coded computation schemes over wireless networks," in *Proc. IEEE Allerton*, Monticello, IL, USA, Jan. 2017, pp. 1256–1263.

[24] M. V. Jamali, M. Soleymani, and H. Mahdavifar, "Coded distributed computing: Performance limits and code designs," in *Proc. IEEE ITW*, Gotland, Sweden, Aug. 2019, pp. 1–5.

[25] P. Peng, E. Soljanin, and P. Whiting, "Diversity vs. parallelism in distributed computing with redundancy," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020, pp. 257–262.

[26] D.-J. Han, J.-Y. Sohn, and J. Moon, "Coded wireless distributed computing with packet losses and retransmissions," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8204–8217, Dec. 2021.

[27] H. A. David and H. N. Nagaraja, *Order Statistics*. Hoboken, NJ, USA: Wiley, 2004.

[28] Y. Zhang, *Monotonic Optimization in Communication and Networking Systems*. Boston, MA, USA: Now Foundations and Trends, 2012.

[29] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded mapreduce," in *Proc. IEEE Allerton*, Monticello, IL, USA, Sep. 2015, pp. 964–971.

[30] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "A scalable framework for wireless distributed computing," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2643–2654, Oct. 2017.

[31] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[32] Y. Yang, M. Interlandi, P. Grover, S. Kar, S. Amizadeh, and M. Weimer, "Coded elastic computing," in *Proc. IEEE ISIT*, Paris, France, Jul. 2019, pp. 2654–2658.

[33] S. Kianidehkordi, N. Ferdinand, and S. C. Draper, "Hierarchical coded matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 67, no. 2, pp. 726–754, Feb. 2021.

[34] N. Liu, K. Li, and M. Tao, "Code design and latency analysis of distributed matrix multiplication with straggling servers in fading channels," *China Commun.*, vol. 18, no. 10, pp. 15–29, Oct. 2021.

[35] K. Li, M. Tao, J. Zhang, and O. Simeone, "Multi-cell mobile edge coded computing: Trading communication and computing for distributed matrix multiplication," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020, pp. 215–220.

[36] L. Qian, W. Wu, W. Lu, Y. Wu, B. Lin, and T. Q. S. Quek, "Secrecy-based energy-efficient mobile edge computing via cooperative non-orthogonal multiple access transmission," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4659–4677, Jul. 2021.

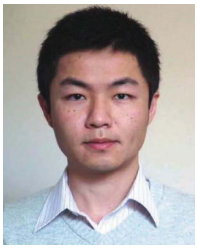
[37] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[38] G. Liang and U. C. Kozat, "FAST CLOUD: Pushing the envelope on delay performance of cloud storage with coding," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 2012–2025, Dec. 2014.

[39] S. Dutta, V. R. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6171–6193, Oct. 2019.

[40] R. Bitar, P. Parag, and S. E. Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4609–4619, Aug. 2020.

[41] N. C. Sagias and G. K. Karagiannidis, "Gaussian class multivariate Weibull distributions: Theory and applications in fading channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3608–3619, Oct. 2005.



Xiaofan He (Senior Member, IEEE) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008, the M.S. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2011, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 2015. He was a tenure-track Assistant Professor of electrical engineering with Lamar University, TX, USA, from 2016 to 2018. He is currently a Faculty Member with the Electronic Information School, Wuhan University, Wuhan. His research interests include wireless communications, networking, computing, edge computing (distributed) and the associated optimization, scheduling, learning, and statistical analysis. Dr. He received the Exemplary Reviewer Award of IEEE TRANSACTIONS ON COMMUNICATIONS in 2014 and 2015, and the Distinguished Member of Technical Program Committee Award of IEEE International Conference on Computer Communications (INFOCOM) in 2018. He is currently serving as an Associate Editor for IEEE ACCESS.

Tianheng Li, photograph and biography not available at the time of publication.

Richeng Jin (Member, IEEE) received the B.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2015, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2020.

He was a Postdoctoral Researcher of electrical and computer engineering at North Carolina State University, from 2021 to 2022. He is currently a Faculty Member with the Department of Information and Communication Engineering, Zhejiang University. His research interests include AI, game theory, and security and privacy in machine learning/artificial intelligence and wireless networks.

Huaiyu Dai (Fellow, IEEE) received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2002.

He was at the Bell Laboratories, Lucent Technologies, Holmdel, NJ, USA, in summer 2000, and at AT and T Laboratories-Research, Middletown, NJ, USA, in summer 2001. He is currently a Professor of electrical and computer engineering with NC State University, Raleigh, holding the title of University Faculty Scholar. His research interests include communications, signal processing, networking, and computing, machine learning and artificial intelligence for communications and networking, multilayer and interdependent networks, dynamic spectrum access and sharing, and security and privacy issues in the above systems.

Dr. Dai was a co-recipient of the Best Paper Awards at 2010 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2016 IEEE Infocom Bigsecurity Workshop, and 2017 IEEE International Conference on Communications. He was an Area Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and the Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. Currently he is a member of the Executive Editorial Committee for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.