Article:
Du, J., Zhao, L., Chu, X. orcid.org/0000-0003-1863-6149 et al. (3 more authors) (2018) Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems. IEEE Transactions on Vehicular Technology. ISSN 0018-9545

https://doi.org/10.1109/TVT.2018.2882991

Enabling Low-Latency Applications in LTE-A Based Mixed Fog/Cloud Computing Systems

Jianbo Du, Liqiang Zhao *Member, IEEE*, Xiaoli Chu *Senior Member, IEEE*, F. Richard Yu *Fellow, IEEE*, Jie Feng, and Chih-Lin I *Senior Member, IEEE*

***Abstract*—In order to enable low-latency computation-intensive applications for mobile user equipments (UEs), computation offloading becomes critical necessary. We tackle the computation offloading problem in a mixed fog and cloud computing system, which is composed of an LTE-A small-cell based fog node, a powerful cloud center, and a group of UEs. The optimization problem is formulated into a mixed-integer non-linear programming (MINLP) problem, and through a joint optimization of offloading decision making, computation resource allocation, resource block (RB) assignment, and power distribution, the maximum delay among all the UEs is minimized. Due to its mixed combinatory, we propose a low-complexity iterative suboptimal algorithm called FAJORA to solve it. In FAJORA, first, offloading decisions are obtained via binary tailored fireworks algorithm (FA); then computation resources are allocated by bisection algorithm. Limited by the uplink LTE-A constraints, we allocate feasible RB patterns instead of RBs, and then distribute power among the RBs of each pattern, where Lagrangian dual decomposition is adopted. Since one UE may be allocated with multiple feasible patterns, we propose a novel heuristic algorithm for each UE to extract the optimal pattern from its allocated patterns. Simulation results verify the convergence of the proposed iterative algorithms, and exhibit significant performance gains could be obtained compared with other algorithms.**

***Index Terms*—Computation offloading, fireworks algorithm, fog computing, LTE-A, resource allocation.**

I. Introduction

With the proliferation of smart user equipments (UEs) and the popularity of low-latency applications [1], the current mobile networks have been pushed to their limits. Mobile cloud computing (MCC) [2] has appeared as a potential way to cope with the above challenges by offloading computations to powerful cloud servers. More recently, fog computing [3] (or mobile edge computing (MEC) [4]) has been put forwarded as an effective complement to MCC and has been deemed as an important paradigm and scenario in 5G [5]. By setting up a virtualized platform between UEs and cloud centers, fog computing can provide computation, storage, and networking services [6], [7] to nearby UEs, and thus to further enhance network performance in energy conservation or delay reduction [5].

Fig.1 shows the typical architecture of a mixed fog/cloud computing system. Utilizing the computation resources of the fog nodes, such as WiFi access points (APs), base stations (BSs), or remote radio heads (RRHs), fog nodes can offer computation services at the edge of the network [3], [4]. Fog nodes can communicate directly with each other, and all the fog nodes are connected to the powerful cloud server through high-speed wired links [3], [4]. The cooperation between the cloud server and the fog nodes can provide users with more efficient and appropriate computation offloading services. However, this new architecture brings many new problems, e.g., how does the fog cooperate with the cloud, i.e., where should computation be offloaded to, and how the resource be allocated, etc., so as to bring the advantages of the new architecture into full play.

In this paper, in order to enable low-latency compute-intensive user applications with fairness among UEs guaranteed, we propose to minimize the maximum delay consumption among all UEs in an LTE-A based mixed fog/cloud computing system by jointly optimizing computation offloading, computation resource allocation, uplink RB assignment and transmit power allocation. Since in the LTE-A uplink, if a UE is assigned with multiple RBs, they must be adjacent RBs [8], [9], so we allocate feasible RB patterns to UEs. Each UE then picks out the optimal pattern from all the assigned feasible RB patterns and then perform power allocation on the RBs of the selected pattern. As the joint optimization problem is a mixed integer non-linear programming (MINLP) problem, we devote to develop low-complexity suboptimal algorithms to decouple it into several subproblems to solve.

The main contributions of this paper are listed as follows.

- We propose a novel general iterative algorithm framework called binary tailored **f**ireworks **a**lgorithm based **j**oint computation **o**ffloading and **r**esource **a**llocation algorithm (FAJORA) to solve the joint optimization problem, where offloading decisions are first decoupled from the rest of the problem and obtained through binary tailored fireworks algorithm.
- We develop a bisection algorithm for computation resource allocation, which is nested in FAJORA.
- We solve uplink RB pattern and power allocation problem, which is still NP-hard, by relaxing , Lagrangian dual

*This work was supported in part by National Natural Science Foundation of China (61771358), National Natural Science Foundation of Shaanxi Province (2018JM6052), Intergovernmental International Cooperation on Science and Technology Innovation (2016YFE0123200), the Fundamental Research Funds for the Central Universities, and the 111 Project (B08038).

J. Du, L. Zhao, J. Feng are with State Key Laboratory of ISN, Xidian University, No.2 Taibainan-lu, Xi'an, 710071, Shaanxi, China. (Email: dujianboo@163.com; lqzhao@mail.xidian.edu.cn; jiefengcl@163.com).

X. Chu is with Department of Electronic and Electrical Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK. (Email: x.chu@sheffield.ac.uk).

F. R. Yu is with the Dept. of Systems and Computer Eng., Carleton University, Ottawa, ON, Canada (e-mail: Richard.Yu@carleton.ca).

C.-L. I is with the Green Communication Research Center, China Mobile Research Institute, Beijing 100053, China. (e-mail: icl@chinamobile.com).

decomposition, and sub-gradient projection methods, to obtain the optimal UE and power allocation for each feasible RB pattern, where each UE may be allocated with multiple feasible patterns.
- We then develop a novel heuristic algorithm to extract the optimal pattern for each UE from all its feasible patterns, taking the exclusiveness required by RB allocation and higher RB utilization into consideration, and thus to obtain more performance gains.

The remainder of this paper is organized as follows. Related works are presented in Section II. Section III introduces the system model and problem formulation. In Section IV, we illustrate the procedure and general structure of FAJORA. The computation resource allocation algorithm is detailed in Section V. In Section VI, we first present the RB pattern and power allocation algorithm, and then detail the heuristic pattern extracting algorithm. Complexity analysis is presented in Section VII. Simulation results are provided in Section VIII. Finally, the paper is concluded in Section IX.

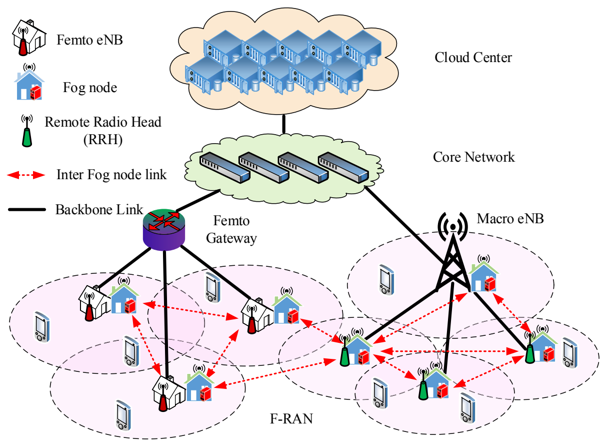


Fig. 1: System architecture of a mixed fog/cloud computing system.

II. Related Works

In single-UE case [10], [11], task partitioning and offloading decision is usually optimized in order to maximize energy savings [10] or to minimize energy consumption [11]. In the most general multi-user scenarios, computation resources and communication resources (e.g., bandwidth, resource blocks, and subcarriers) are shared among UEs. Therefore, except for offloading decisions, resource allocation is another important issue needs to be investigated. In [12], game theory was utilized in an MCC environment. According to other UEs' decisions, each UE optimized its offloading decision and thus to minimize its weighted cost. In [13], offloading decisions were optimized for all UEs to minimize the network energy consumption in an MCC system. The authors in [14] investigated transmit power optimization under given offloading decisions, in order to minimize the system energy consumption. The formulation in [15] combined task level offloading decision optimization and transmit power allocation in multiuser MCC and MEC scenarios, respectively, to minimize a weighted system cost of delay and energy consumption. In [16], except for optimizing offloading decisions and transmit power allocation, the authors extended computation resource allocation into the optimization framework to further reduce latency and energy consumption of all the UEs in an MEC network. The following references optimized the allocation of other forms of radio resources instead of transmit power. The authors in [17] formulated a joint optimization of the offloading decision making, resource block (RB) allocation, and computation resource allocation in the MEC server, with transmit power given as a constant, to minimize the total weighted cost of delay and energy consumption of all UEs. In [18], in order to minimize system energy consumption, the authors performed a joint optimization of computation offloading, subcarrier assignment, and computation resource allocation in a fog computing system. The authors in [19] formulated a system energy consumption minimization problem with the required delay tolerance satisfied in an MCC system, by a joint optimization of beamformer designing, computation resources allocation and offloading decision making. The authors in [20] first proposed to jointly optimize the offloading decision making, computation resource allocation, and radio transmit rate allocation pioneeringly in an MCC system, in order to conserve energy while satisfying user delay constraints, while radio resources were allocated in a coarse-grained unit of bit/s.

To summarize, [12], [13] only optimized offloading decisions, [14] only optimized transmit power allocation, [15] combined the two aspects for further optimization, and [16]–[19] integrated computation resource allocation into the optimization framework besides offloading decision optimization and radio resource allocation. However, in [14]–[20], radio resource allocation only covered a certain dimension of radio resources, such as transmit power, RB, or subcarrier allocation, without a joint optimization of multidimensional wireless resources for further performance gains. What's more, applications were offloaded either to the cloud server [12]–[15], [19], [20] or to the fog node [16]–[18], without a cooperation between the both for providing much stronger offloading services. Moreover, the above related works [13]–[20] concerned the system-level performance, without considering that of individual UEs. Consequently, UEs with higher transmit rate will benefit from computation offloading, but at the expense of a performance decline of the UEs with lower transmit rate, giving rise to unfairness among UEs.

III. System Model and Problem Formulation

In this section, we first describe the concerned scenario, then discuss the delay consumption in local, fog, and cloud processing modes, respectively, and finally we formulate our optimization problem under the concerned scenario.

A. Description of the Concerned Scenario

We consider a system comprising N UEs, an LTE-A small cell based fog node, and a distant cloud server. The fog node and the cloud are jointed by a fiber link, while all the N UEs are connected to the fog node through wireless links sharing K RBs in each transmission time interval (TTI, occupying 1 ms [8], [9]). Each RB is allocated to at most one UE in the small cell [8], [9], [21]. We consider a quasi-static scene where all UEs and the wireless channels keep still within an offloading period (usually several seconds [12], containing several thousands of TTIs). The assumption holds for many actual applications such as face recognition, natural language processing, and so on, where the input is not so large that computation offloading could be accomplished within a short time less than the time duration of UEs' mobility and wireless channels' variation. Thus, in the following, we consider the offloading period as the time unit where the optimization is performed [11], [12], [14], [16], [22], [23], and all the TTIs in the same offloading period adopt the same optimization results.

Each UE has only one inseparable application may be executed locally or remotely in application-level through the following process. Firstly, it sends an offloading request (including the information about the application and the UE itself) to the manager in the fog node [23]. The manager collects the information about wireless channel states and the available resources in the fog node, together with the offloading requests, it determines the offloading decision (where the application be processed, i.e., in the UE locally, in the fog, or in the cloud) and the associated resource allocation for each UE. The offloading decisions are then sent back to all the UEs, and the corresponding resources will be allocated to them in offloading. As an offloading request is usually very tiny, we suppose that no buffer is needed for queueing the computation requests [22]. Also, the delay in decision making is not considered to enable tractable analysis [23].

Denote the UE set as $\mathcal{N}$, and the offloading decision of UE$_n$ as x_n, y_n, z_n. Let $x_n = 1, y_n = 1, z_n = 1$ represent that the application is processed by UE$_n$ itself, by the fog node, or by the cloud, respectively; otherwise, $x_n = 0, y_n = 0, z_n = 0$. Consequently, we have

$$x_n + y_n + z_n = 1, \ \forall n \in \mathcal{N}. \tag{1}$$

The offloading decisions of all UEs is collected in the offloading matrix $\mathbf{\Pi}$, which is given by $\mathbf{\Pi} = \begin{bmatrix} x_1, & ..., & x_N \\ y_1, & ..., & y_N \\ z_1, & ..., & z_N \end{bmatrix}_{3\times N}$, where the nth column is the offloading decision of UE$_n$.

The fog node has the ability to process applications, subjecting to its computation capability. When multiple UEs choose fog-processing, the fog node will allocate computation resources (in CPU cycles/s) to them. When cloud-executing is selected by multiple UEs, their applications will firstly be transmitted from the UEs to the fog node through a shared wireless access link, and then be forwarded by the fog node to the cloud server through a wired fiber link. Since the computation resources in the cloud server is sufficient, and the capacity of the wired link is large enough, the allocation of those resources (the decisions are made at the cloud server) will not be discussed and these resources allocated to each UE is given as known quantities [20]. However, the limited radio resource needs to be assigned among all the remote-executing UEs (including all the UEs with $y_n = 1$ or $z_n = 1$). Since the output of remote processing is usually very tiny, only the uplink is discussed [11], [12], [16].

The application of UE$_n$ can be denoted as $\Lambda_n = \{D_n, \lambda_n\}, \ n \in \mathcal{N}$, where D_n represents the input data size (in bits), and λ_n denotes processing density or computation complexity (in CPU cycles/bit) of the application [1]. The number of CPU cycles C_n(aka.computation load) required to complete executing the application is modeled as $C_n = D_n\lambda_n$, which increases with both the input data D_n and the processing density λ_n. Using program profilers [10], [16], the manager can obtain D_n, C_n and λ_n beforehand easily. For each UE there is a clone in the fog node, and the program of Λ_n is backed up in the clone [6], [7], [11], [23], and can be downloaded easily by the cloud server through the wired link [10], [14], noting that the overhead for setting up and synchronize the clone is neglected similar to many existing works [6], [7], [11], [23]. Hence, only the input data with size D_n bits will be transmitted from UE$_n$ when offloading.

Feasible RB allocation pattern matrix $\mathbf{W}^n = \{w^n_{k,j}\}$ of size $K \times J$ can be constructed for each UE$_n$, where $w^n_{k,j} = 1$ indicates that RB k is allocated to UE$_n$ in the jth pattern, otherwise $w^n_{k,j} = 0$. For example, assuming there are $K = 4$ RBs, the feasible pattern matrix for UE$_n$ is given by

$$\mathbf{W}^n = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{K\times J}, n \in \mathcal{N}, \tag{2}$$

where J is the total number of feasible patterns given by $J = \frac{1}{2} \times (K^2 + K) + 1$ [8]. The uplink RB pattern allocation matrix is formed as $\mathbf{S} = \{s_{n,j}\}_{N\times J}$, where the binary variable $s_{n,j} = 1$ indicates that UE$_n$ selects pattern j, and otherwise $s_{n,j} = 0$.

If pattern j is assigned to UE$_n$, using Shannon's formula, the maximum achievable transmit rate for UE$_n$ on RB k in pattern j can be expressed as

$$\begin{aligned} r_{n,k(j)} &= W_0 \log_2 \left(1 + \frac{p_{n,k(j)} h_{n,k(j)}}{\sigma^2}\right) \\ &= W_0 \log_2 \left(1 + p_{n,k(j)} g_{n,k(j)}\right), \end{aligned} \tag{3}$$

where W_0 is the bandwidth of an RB (in Hz); $p_{n,k(j)}$ is the transmission power of UE$_n$ on RB k in pattern j; $h_{n,k(j)}$ is channel gain, which is assumed to be known at UE$_n$ and remain constant but may change at the boundary of each offloading period [21]; $g_{n,k(j)} = \frac{h_{n,k}}{\sigma^2}$, and σ^2 is the power of additive white Gaussian noise. Thus, the maximum achievable transmit rate and power of UE$_n$ on pattern j are given by

$$r_{n,j} = \sum_{k=1}^{K} r_{n,k(j)}, \tag{4}$$

$$p_{n,j} = \sum_{k=1}^{K} p_{n,k(j)}, \tag{5}$$

and the total achievable transmit rate and power of UE$_n$ are given by

$$r_n = \sum_{j=1}^{J} s_{n,j} r_{n,j} = \sum_{j=1}^{J}\sum_{k=1}^{K} s_{n,j} r_{n,k(j)}, \tag{6}$$

$$p_n = \sum_{j=1}^{J} s_{n,j} p_{n,j} = \sum_{j=1}^{J}\sum_{k=1}^{K} s_{n,j} p_{n,k(j)}, \tag{7}$$

where p_n should be less than the maximum transmit power p_n^{max} of UE$_n$. It should be noted that in equations (3), (4), and (6), we assume the wireless channel gain of each RB to be a random value, without specifically considering the impact of the number of antennas.

In the following we will discuss the delay consumption of local, fog and cloud processing, respectively.

B. Delay Under Different Offloading Scenarios

Denote the local processing capability (in CPU cycles/s) of UE$_n$ as f_n^{loc}, then the delay of local processing is

$$T_n^{loc} = \frac{C_n}{f_n^{loc}}. \tag{8}$$

If fog-processing is selected for UE$_n$, then, it needs to transmit the input data of size D_n to the fog node. Assume fog processing only starts when all the input data has been received, and denote the computation resource allocated to UE$_n$ by f_n^{fog} (in CPU cycles/s), then the delay of UE$_n$ under fog processing is given by

$$T_n^{fog} = \frac{D_n}{r_n} + \frac{C_n}{f_n^{fog}}. \tag{9}$$

If the application of UE$_n$ is offloaded to the cloud server, the input data is first transmitted to the fog node, and then sent to the cloud server by the fog node. Given the rate of the high-speed wired fiber link and the cloud processing capability for UE$_n$ as R_n^{fc} (in bit/s) and f_n^c (in CPU cycles/s), respectively, then the wired transmit delay and processing delay in the cloud are given by $T_n^{fc} = D_n/R_n^{fc}$ and $T_n^c = C_n/f_n^c$, respectively. Thus the total delay of UE$_n$ for cloud processing is given by

$$T_n^{cloud} = \frac{D_n}{r_n} + T_n^{fc} + T_n^c. \tag{10}$$

A summary of the mainly used notations are presented in TABLE I.

C. Problem Formulation

We then formulate the joint optimization of computation offloading and resource allocation and show its NP-hardness.

Based on (8)-(10), the delay of UE$_n$ is given by

$$T_n = T_n^{loc} x_n + T_n^{fog} y_n + T_n^{cloud} z_n. \tag{11}$$

We intend to minimize the maximum delay consumption among all UEs, by jointly optimizing the offloading decision $\mathbf{\Pi}$, the RB pattern allocation $\mathbf{S} = \{s_{n,j}\}_{N\times J}$, the transmit power allocation $\mathbf{P} = \{p_{n,k(j)}\}_{N\times K\times J}$, and the computation resource assignment $\mathbf{f}^{fog} = \{f_1^{fog}, ..., f_N^{fog}\}$. The computation resources in the fog node will be allocated only to the fog-executing UEs, the set of which is denoted as $\mathcal{N}_1$. RBs will be assigned among all the remote-processing (including fog-processing and cloud-processing) UEs, the set of which is denoted as $\mathcal{N}_2$. The joint optimization problem is formulated as

$$\begin{aligned}
(\mathcal{P}_1):\ & \min_{\mathbf{\Pi},\mathbf{f}^{fog},\mathbf{S},\mathbf{P}} \ \max_{n\in\mathcal{N}} \ T_n \qquad (12)\\
\text{s.t. } & (\text{C1}):\ x_n, y_n, z_n \in \{0,1\}, \quad \forall n\in\mathcal{N},\\
& (\text{C2}):\ x_n + y_n + z_n = 1, \quad \forall n\in\mathcal{N},\\
& (\text{C3}):\ \sum_{n\in\mathcal{N}_1} f_n^{fog} \le F^{fog},\\
& (\text{C4}):\ f_n^{fog} \ge 0, \quad \forall n\in\mathcal{N}_1,\\
& (\text{C5}):\ s_{n,j}\in\{0,1\},\ \forall j\in\mathcal{J},\ \forall n\in\mathcal{N}_2,\\
& (\text{C6}):\ \sum_{n\in\mathcal{N}_2} s_{n,j} = 1,\ \forall j\in\mathcal{J}|\{j=1\},\\
& (\text{C7}):\ \sum_{j\in\mathcal{J}} s_{n,j} = 1, \quad \forall n\in\mathcal{N}_2,\\
& (\text{C8}):\ \sum_{n\in\mathcal{N}_2}\sum_{j\in\mathcal{J}} s_{n,j}\mathbf{W}^n \le 1,\ \forall k\in\mathcal{K},\\
& (\text{C9}):\ p_{n,k(j)} \ge 0,\ \forall n\in\mathcal{N}_2, \forall k\in\mathcal{K}, \forall j\in\mathcal{J},\\
& (\text{C10}):\ \sum_{j\in\mathcal{J}} s_{n,j}p_{n,j} \le p_n^{max}, \quad \forall n\in\mathcal{N}_2.
\end{aligned}$$

TABLE I: Notation Definitions

Symbol	Definition
T_n^{loc}, T_n^{fog}, T_n^{cloud}	Delay of UE$_n$ in local/fog/cloud processing
f_n^{loc}, f_n^{fog}, f_n^c	Processing ability of UE$_n$ in local/fog/cloud processing
D_n,C_n,λ_n	Data size/computation load/processing density of the application of UE$_n$
F^{fog}	Total computation capability of the fog node
p_n^{max}	The maximum transmit power of UE$_n$
$g_{n,k(j)}$	Power gain of UE$_n$ on RB k in pattern j
$s_{n,j}$	Indicator whether pattern j is allocated to UE$_n$
r_n, p_n	Wireless transmit rate/power of UE$_n$
R_n^{fc}	Wired link rate of UE$_n$ between fog and cloud
$r_{n,j}, p_{n,j}$	Transmit rate/power of UE$_n$ on pattern j
$r_{n,k(j)}$ $p_{n,k(j)}$	Transmit rate/power of UE$_n$ on RB k in pattern j
$\mathbf{S}$,$\mathbf{P}$	RB Pattern/power allocation matrix
x_n, y_n, z_n	Offloading decisions of UE$_n$
$\mathbf{\Pi}$	Matrix of all UEs' offloading decisions
$\mathcal{N}$,N	The set/number of all UEs
$\mathcal{N}_1$,N_1	The set/number of fog-executing UEs
$\mathcal{N}_2$,N_2	The set/number of remote-executing UEs
$\mathcal{K}$, K	The set/number of RBs
$\mathcal{J}$,J	The set/number of all feasible patterns
$\mathbf{W}^n$	Feasible RB pattern matrix for UE$_n$
I,M,γ	Number of fireworks/total explosion sparks/ mutation sparks
L	Number of iterations of FA

In problem $(\mathcal{P}_1)$, (C1) and (C2) give each UE$_n$ the restraints on its offloading decisions; (C3) shows that the total allocated computation resource should be less than the total computation capability F^{fog} in the fog node; (C4) indicates that the computation resource allocated to each fog-processing UE$_n$ should be nonnegative; (C5) is the binary constraint of RB pattern allocation; (C6) indicates that except for pattern 1, each pattern can be assigned to only one UE, while pattern 1 can be assigned to any UE that is not allocated with RB; (C7) requires that each UE can only be assigned with one pattern; (C8) guarantees that each RB is allocated to only one UE; (C9) requires the power on each RB is nonnegative; (C10) is the maximum transmit power constraint of each UE.

Proposition 1: Problem $(\mathcal{P}_1)$ is NP-hard.

Proof: See Appendix A. ■

IV. Binary Tailored Fireworks Algorithm Based Joint Computation Offloading and Resource Allocation Algorithm

Since problem $(\mathcal{P}_1)$ is hard to solve, in this section, we decouple it into two subproblems, i.e., offloading decisions making and resource allocation, which are solved by the proposed algorithm FAJORA.

When a firework is ignited, a burst of sparks will fill the surrounding space around the firework. Inspired by the phenomenon, the authors in [24] proposed a new kind of heuristic algorithm called fireworks algorithm (FA), and have verified that it performs well in convergence speed and global searching, compared with other heuristic algorithms such as genetic algorithm (GA) [25] and particle swarm optimization algorithm (PSO) [26]. As good candidates for solving MINLP problem, heuristic algorithms have been used widely in many fields such as radio resource allocation [26] and fuzzy control [27], etc. However, to the best understanding of us, seldom have they been used efficiently in computation offloading, especially fireworks algorithm. In the following we will introduce Binary Tailored Fireworks Algorithm(BTFA) firstly, then we propose the general framework of our proposed FAJORA for the considered scenario.

A. Some Concepts

1) Fireworks and Sparks: Fireworks and the newly generated sparks represent feasible solutions in the solution space. Specifically, a firework/spark indicates an offloading decision matrix $\mathbf{\Pi}$ in the considered problem.

2) Fitness Function and Fitness Value: Fitness values are employed to evaluate the performance of feasible solutions. We take the objective function in $(\mathcal{P}_1)$ as the fitness function to obtain the fitness value of each firework.

3) Binary Matrix Distance: Binary matrix distance means the Manhattan distance of two binary matrixes, i.e., the sum of the distance between each element of the two matrixes. Suppose two matrixes X and Y are $m \times n$-dimensional, the distance between the two matrixes is

$$d(X,Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} |X_{i,j} - Y_{i,j}|. \tag{13}$$

B. Overview of FA

The typical steps for solving a problem with FA can be summarized as follows: First, initialize a swarm of fireworks and obtain their fitness values according to the specified fitness function. Then each firework performs explosion operator to generate some explosion sparks around the firework within a certain amplitude. The number of sparks and the explosion amplitude are obtained according to the fitness value of each firework. The better fitness value of a firework, the more explosion sparks it will produce, and the smaller amplitude will be, and vice versa. After that several Gaussian mutation sparks are generated in order to keep population diversity. Afterwards, from the population of fireworks and sparks, several individuals are picked out as the fireworks of the next iteration. The procedure of explosion, mutation and selection is repeated until the algorithm reaches convergence, or reaches the maximum iteration index. Finally, from the individuals obtained in the last iteration, the individual with the best fitness value is selected as the solution to the constructed problem.

C. Operations of BTFA

Given total I fireworks, the major operations of the binary tailored FA are listed as follows.

1) Explosion: The number of the explosion sparks for the ith firework $\mathbf{\Pi}_i$ is given by

$$\chi_i = ceil\left(M \frac{f_{max} - f(\mathbf{\Pi}_i) + \epsilon}{\sum\limits_{i=1}^{I}(f_{max} - f(\mathbf{\Pi}_i)) + \epsilon} \right), \tag{14}$$

where $ceil(\cdot)$ is round up function, M is a parameter constraining the total number of explosion sparks, $f_{max} = max(f(\mathbf{\Pi}_i)), i = 1, ..., I$ is the worst fitness value among all the I fireworks, and ϵ is an extremely tiny number which is used to avoid zero-division-error.

To avoid that one firework may generate too less or too many explosion sparks, bounds are defined for each χ_i, which is given by

$$\hat{\chi_i} = \begin{cases} round(aM), & if\ \chi_i < aM \\ round(bM), & if\ \chi_i > bM, a < b < 1, \\ round(s_i), & otherwise \end{cases} \tag{15}$$

where $round(\cdot)$ is the rounding off function, a and b are given constants, and $\hat{\chi_i}$ is the number of actual generated explosion sparks.

For firework $\mathbf{\Pi}_i$, each explosion spark is generated like this: 1) choose β columns from the N columns randomly; 2) perform cyclic shift on each selected column; 3) the rest columns are kept unchanged.

2) Mutation: To improve the spark diversity and thereby to increase searching capability, mutation is introduced. From the I fireworks, we choose γ of them randomly, each of which will generate a mutation spark like this: 1) choose β columns from the N columns randomly; 2) for each of the selected column, reset it with a random feasible offloading decision; 3) the rest columns are kept unchanged.

3) Selection: After explosion and mutation, there exist three different kinds of individuals, i.e., fireworks, explosion sparks, and mutation sparks. The individual with the best fitness value is always kept as the first firework of the next generation. To keep diversity of the population, other $I-1$ fireworks are selected form the rest individuals according to their selected probabilities, which are obtained from their binary matrix distance to all other individuals as follows.

According to (13), the binary matrix distance between an individual $\mathbf{\Pi}_i$ and other individuals is given by

$$R(\mathbf{\Pi}_i) = \sum_{j \in \mathcal{K}} d(\mathbf{\Pi}_i, \mathbf{\Pi}_j), \tag{16}$$

where $\mathcal{K}$ is the population of all current individuals including both fireworks and sparks.

Consequently, the selected probability of individual $\mathbf{\Pi}_i$ is given by

$$p(\mathbf{\Pi}_i) = \frac{R(\mathbf{\Pi}_i)}{\sum\limits_{j \in \mathcal{K}} R(\mathbf{\Pi}_i)}. \tag{17}$$

*D. BTF***A** *Based* ***J****oint Computation* ***O****ffloading and* ***R****esource* ***A****llocation Algorithm (FAJORA)*

Applying the proposed binary tailored operators to traditional fireworks algorithm, and adopting the same procedure with FA, we obtain BTFA. Using BTFA we can solve the original formulated problem $(\mathcal{P}_\mathbf{1})$, i.e., the final offloading decision $\mathbf{\Pi}$ and the corresponding resource allocation can be obtained. Detailed algorithm framework is called **BTFA** based **j**oint Computation **o**ffloading and **r**esource **a**llocation algorithm (FAJORA) and is summarized in Algorithm 1 as folllows.

Algorithm 1 FAJORA

```
Initialization:
 1: Set $N$, $K$, $F^{fog}$, $I$, $M$, $\beta$, $\gamma$, $a$ and $b$.
 2: Initialize $D_n, C_n, f_n^{loc}, p_n^{max}, R_n^{fc}, f_n^c$ of each UE.
 3: Generate $I$ random fireworks $\mathbf{\Pi}_1, ..., \mathbf{\Pi}_I$ in the searching
    space, perform resource allocation under each firework,
    and obtain the fitness value of each firework.
Iteration:
 4: while 1 or $l <= L$ do
 5:   for $i <= I$ do
 6:     Obtain the number of explosion sparks $\chi_i$ according
        to (14) and (15).
 7:     for $p <= \chi_i$ do
 8:       Perform explosion to generate explosion spark $p$.
 9:       Perform resource allocation under $p$.
10:       Calculate the fitness value of $p$.
11:     end for
12:   end for
13:   for $j <= \gamma$ do
14:     Generate mutation spark $j$.
15:     Perform resource allocation under $j$.
16:     Obtain the fitness value of $j$.
17:   end for
18:   The best individual is considered as the first firework
      of the next iteration, and the other $I - 1$ fireworks
      are chosen from the rest individuals according to the
      selected probability in (17).
19: end while
20: Among the fireworks selected in the last iteration, the one
    with the minimum fitness value is considered as $\mathbf{\Pi}^*$.
21: Output: $\mathbf{\Pi}^*$ and corresponding resource allocation.
```

V. Computation Resource Allocation

Next we will tackle the resource allocation subproblem embedded in Steps 3, 9, and 15 in Algorithm 1, where both radio and computation resource allocation need to be determined. In this section, we describe how to obtain the computation resource allocation, while radio resource allocation will be presented in the next section.

After offloading decision $\mathbf{\Pi}$ has been obtained, problem $(\mathcal{P}_\mathbf{1})$ degrades to the joint optimization of allocating computation resources, RB patterns, and transmit power as follows

$$(\mathcal{P}_\mathbf{2}): \min_{\mathbf{f}^{fog}, \mathbf{S}, \mathbf{P}} \max_{n \in \mathcal{N}} T_n \tag{18}$$
$$\text{s.t.} \ \ (\text{C3}) - (\text{C10}).$$

To reduce the computation complexity, we divide $(\mathcal{P}_\mathbf{2})$ into two subproblems: computation resource allocation and radio resource assignment. For computation resource allocation among UEs in $\mathcal{N}_1$, assuming the radio resource assignment $\mathbf{S}$ and $\mathbf{P}$ are given, we have

$$(\mathcal{P}_\mathbf{3}): \min_{\mathbf{f}^{fog}} \max_{n \in \mathcal{N}_1} \left(\frac{C_n y_n}{f_n^{fog}} + B_n \right) \tag{19}$$
$$\text{s.t.} \ \ (\text{C3}), (\text{C4}),$$

where $B_n = \frac{C_n x_n}{f_n^{loc}} + (T_n^{fc} + T_n^c) z_n + \frac{D_n(y_n + z_n)}{r_n} = \frac{D_n}{r_n}, n \in \mathcal{N}_1$ is a constant. Letting $\frac{C_n y_n}{f_n^{fog}} + B_n = \frac{C_n}{f_n^{fog}} + B_n \leq \tau, n \in \mathcal{N}_1$, the non-smooth problem $(\mathcal{P}_\mathbf{3})$ is converted to

$$(\mathcal{P}_\mathbf{4}): \min_{\mathbf{f}^{fog}, \tau} \tau \tag{20}$$
$$\text{s.t.} \ \ (\text{C3}), \ (\text{C4}),$$
$$(\text{C16}): \ \frac{C_n}{f_n^{fog}} + B_n \leq \tau, \ \forall n \in \mathcal{N}_1.$$

Since $\frac{C_n}{f_n^{fog}} \geq 0$, then $\tau - B_n \geq 0$, so we have

$$0 \leq \frac{C_n}{\tau - B_n} \leq f_n^{fog}, \ \forall n \in \mathcal{N}_1. \tag{21}$$

Consequently, we have

$$\sum_{n \in \mathcal{N}_1} \frac{C_n}{\tau - B_n} \leq \sum_{n \in \mathcal{N}_1} f_n^{fog} \leq F^{fog}. \tag{22}$$

In order to minimize the maximum delay among all the fog-executing UEs, the UE with the maximum delay (which is denoted by UE*) needs to be allocated with more computation resources, so less computation resources will be left for all the other UEs (i.e., UEs in the set $\mathcal{N}_1 | UE^*$), leading to an increase in the delay of those UEs. Performing the above process iteratively, all the computation resources will be distributed evenly among all the fog-executing UEs in the end. Thus we have

$$\sum_{n \in \mathcal{N}_1} \frac{C_n}{\tau - B_n} = \sum_{n \in \mathcal{N}_1} f_n^{fog} = F^{fog}. \tag{23}$$

Thus problem $(\mathcal{P}_\mathbf{4})$ can be converted to

$$(\mathcal{P}_\mathbf{5}): \min_{\tau} \tau \tag{24}$$
$$\text{s.t.} \ \ (\text{C17}): \ \sum_{n \in \mathcal{N}_1} \frac{C_n}{\tau - B_n} = F^{fog}.$$

Since the left-hand side of (C17) is a monotonic decreasing function about τ, bisection method can be used to solve problem $(\mathcal{P}_\mathbf{5})$, which is detailed in Algorithm 2.

Algorithm 2 Bisection Method for Computation Resource Allocation

```
Initialization:
 1: Set $\tau^{min} = max\{B_n\}$, $\tau^{max} = \sum_{n\in\mathcal{N}_1}\left(\frac{C_n N_1}{F^{fog}} + B_n\right)$.
 2: Set $i = 1$ and the precision $\varepsilon > 0$.
Iteration:
 3: while 1 do
 4:     $\tau^i = (\tau^{min} + \tau^{max})/2$.
 5:     if $\mid \tau^{max} - \tau^{min} \mid \le \varepsilon$ then
 6:         $\tau^* = \tau^i$.
 7:     else
 8:         if $\sum_{n\in\mathcal{N}_1} \frac{C_n}{\tau^i - B_n} > F^{fog}$ then
 9:             $\tau^{min} = \tau^i$.
10:         else
11:             $\tau^{max} = \tau^i$.
12:         end if
13:     end if
14:     $i = i + 1$.
15: end while
16: Let $\tau^* = \tau^i$ and substituting it into (23), $\mathbf{f}^{fog^*}$ is obtained.
17: Output: $\mathbf{f}^{fog^*}$.
```

VI. Communication Resource Assignment

After computational resource allocation is obtained, problem $(\mathcal{P}_\mathbf{2})$ degrades to the joint optimization of RB pattern assignment and power allocation among all remote-executing UEs in $\mathcal{N}_2$. Denoting the RB pattern assignment and transmit power allocation as $\mathbf{S} = \{s_{n,j}\}_{N_2\times J}$ and $\mathbf{P} = \{p_{n,k(j)}\}_{N_2\times K\times J}$, then the radio resource allocation subproblem is given by:

$$(\mathcal{P}_\mathbf{6}): \min_{\mathbf{S},\mathbf{P}} \max_{n\in\mathcal{N}_2} \left(\frac{D_n(y_n+z_n)}{r_n} + V_n\right) \quad (25)$$
$$\text{s.t.} \quad (\text{C5}) - (\text{C10}),$$

where $V_n = \frac{C_n y_n}{f_n^{fog}} + \frac{C_n x_n}{f_n^{loc}} + (T_n^{fc} + T_n^c)z_n = \frac{C_n y_n}{f_n^{fog}} + (T_n^{fc} + T_n^c)z_n$ is a constant. However, $(\mathcal{P}_\mathbf{6})$ is still an NP-hard problem as proven in Appendix A. To make the problem tractable, we first relax each $s_{n,j}$ to a continuous interval, i.e., $0 \le s_{n,j} \le 1$; then we define a new matrix $\mathbf{\Phi} = \{\phi_{n,k(j)}\}_{N_2\times K\times J} = \{s_{n,j}p_{n,k(j)}\}_{N_2\times K\times J}$ to replace $\mathbf{P} = \{p_{n,k(j)}\}_{N_2\times K\times J}$. Noting $\frac{D_n(y_n+z_n)}{r_n} = \frac{D_n}{r_n}, n \in \mathcal{N}_2$, and letting $\max_{n\in\mathcal{N}_2}\left(\frac{D_n}{r_n} + V_n\right) = \tau_1$, we have $\frac{D_n}{r_n} + V_n \le \tau_1, n \in \mathcal{N}_2$, then $(\mathcal{P}_\mathbf{6})$ can be rearranged as

$$(\mathcal{P}_\mathbf{7}): \min_{\mathbf{S},\mathbf{\Phi},\tau_1} \tau_1 \quad (26)$$
$$\text{s.t.} \quad (\text{C5}): \ 0 \le s_{n,j} \le 1, \ \ \forall n \in \mathcal{N}_2, \ \forall j \in \mathcal{J},$$
$$(\text{C6}) - (\text{C8}),$$
$$(\text{C9}): \ \phi_{n,k(j)} \ge 0, \quad \forall n \in \mathcal{N}_2, \ \forall k \in \mathcal{K},$$
$$(\text{C10}): \ \sum_{j=1}^{J}\sum_{k=1}^{K} \phi_{n,k(j)} \le p_n^{max}, \ \forall n \in \mathcal{N}_2,$$
$$(\text{C18}): r_n \ge \frac{D_n}{\tau_1 - V_n}, \qquad \forall n \in \mathcal{N}_2.$$

Proposition 2: Problem $(\mathcal{P}_\mathbf{7})$ is jointly convex in $\mathbf{S}$ and $\mathbf{\Phi}$ for given τ_1.

Proof: See Appendix B. ■

Since $(\mathcal{P}_\mathbf{7})$ is convex, Slater's condition [28] is met and 0 duality gap can be assured, so we can solve it employing Lagrangian dual decomposition and sub-gradient projection method [29]. Once the optimal solution $\{\mathbf{S}^*, \mathbf{\Phi}^*\}$ to $(\mathcal{P}_\mathbf{7})$ is obtained, the optimal solution $\{\mathbf{S}^*, \mathbf{P}^*\}$ to $(\mathcal{P}_\mathbf{6})$ is obtained.

A. Lagrange Dual Decomposition Based RB Pattern and Power Allocation

To reduce the number of dual variables and thus to improve convergence speed, the partial Lagrange function of $(\mathcal{P}_\mathbf{7})$ is given by (27), where $\boldsymbol{\mu} = \{\mu_n\} \succeq 0, n \in \mathcal{N}_2$ and $\boldsymbol{\omega} = \{\omega_n\} \succeq 0, n \in \mathcal{N}_2$ are Lagrange dual variables corresponding to (C10) and (C18) in $(\mathcal{P}_\mathbf{7})$, respectively.

The Lagrange dual function is given by

$$D(\boldsymbol{\mu},\boldsymbol{\omega}) = \min_{\mathbf{S},\mathbf{\Phi},\tau_1\in\{(\text{C5})-(\text{C9})\}} L(\mathbf{S},\mathbf{\Phi},\tau_1,\boldsymbol{\mu},\boldsymbol{\omega}), \quad (28)$$

which can be decomposed into $J-1$ independent subproblems (except for pattern $j = 1$). The jth subproblem under given dual variables $(\boldsymbol{\mu},\boldsymbol{\omega})$ is given as

$$(\mathcal{P}_\mathbf{8}): \min_{\mathbf{s}_j,\mathbf{\Phi}_j,\tau_1} L_j(\mathbf{s}_j,\mathbf{\Phi}_j,\tau_1) \quad (29)$$
$$\text{s.t.} \quad (\text{C5}) - (\text{C9}),$$

where $\mathbf{s}_j = \{s_{n,j}\}^T_{N_2 * 1}$, $\mathbf{\Phi}_j = \{\phi_{n,k(j)}\}_{N_2\times K}$ is the sub-matrix of $\mathbf{S}$ and $\mathbf{\Phi}$ for RB pattern j, and

$$L_j(\mathbf{s}_j,\mathbf{\Phi}_j,\tau_1) = \tau_1 - \sum_{n\in\mathcal{N}_2}\mu_n\sum_{k=1}^{K}\phi_{n,k(j)} \quad (30)$$
$$+ \sum_{n\in\mathcal{N}_2}\omega_n\sum_{k=1}^{K}s_{n,j}W_0\log_2\left(1+\frac{\phi_{n,k(j)}g_{n,k(j)}}{s_{n,j}}\right).$$

From $(\mathcal{P}_\mathbf{8})$ we know that $\mathbf{s}_j$ contains only one nonzero binary entry, because every pattern j can only be allocated to one UE as required in constraint (C6). For pattern j, assuming $s_{n,j}, n \in \mathcal{N}_2$, is known, we optimize power allocation for each RB in pattern j. Let

$$\xi_{n,j} = \omega_n\sum_{k=1}^{K}s_{n,j}W_0\log_2\left(1+\frac{\phi_{n,k(j)}g_{n,k(j)}}{s_{n,j}}\right)$$
$$-\sum_{k=1}^{K}\mu_n\phi_{n,k(j)} + \tau_1, \quad (31)$$

$$L(\mathbf{S}, \boldsymbol{\Phi}, \tau_1, \boldsymbol{\mu}, \boldsymbol{\omega}) \tag{27}$$
$$= \tau_1 + \sum_{n\in\mathcal{N}_2} \mu_n \left(p_n^{max} - \sum_{j=1}^{J}\sum_{k=1}^{K} \phi_{n,k(j)} \right) + \sum_{n\in\mathcal{N}_2} \omega_n \left[\sum_{j=1}^{J}\sum_{k=1}^{K} s_{n,j} W_0 \log_2 \left(1 + \frac{\phi_{n,k(j)} g_{n,k(j)}}{s_{n,j}} \right) - \frac{D_n(y_n+z_n)}{\tau_1 - V_n} \right].$$

$(\mathcal{P}_\mathbf{8})$ reduces to the following problem

$$(\mathcal{P}_\mathbf{9}):\ \Gamma_{n,j} = \min_{\phi_{n,k(j)}} \xi_{n,j} \tag{32}$$
$$\text{s.t. } (\text{C8}'):\ \phi_{n,k(j)} \geq 0,\ \forall n \in \mathcal{N}_2,\ k \in \mathcal{K}.$$

Let $\frac{\partial \xi_{n,j}}{\partial \phi_{n,k(j)}} = 0$, the optimal power allocation for each RB in pattern j is obtained as follows

$$p_{n,k(j)}^* = \frac{\phi_{n,k(j)}}{s_{n,j}} = \left(\frac{\omega_n W_0}{\mu_n \ln 2} - \frac{1}{g_{n,k(j)}} \right)^+, \tag{33}$$

where $x^+ \triangleq max\{0, x\}$. By substituting $p_{n,k(j)}^*$ in place of $\frac{\phi_{n,k(j)}}{s_{n,j}}$ in (31), we obtain $\Gamma_{n,j}$ as

$$\Gamma_{n,j} = \left(\xi_{n,j} \mid p_{n,k(j)} = p_{n,k(j)}^* \right). \tag{34}$$

Performing the procedure (31)-(34) for each UE in $\mathcal{N}_2$, we obtain $\Gamma_j = \{\Gamma_{n,j}\}, n \in \mathcal{N}_2$. The UE$_n$ with the minimum $\Gamma_{n,j}, n \in \mathcal{N}_2$, is selected as the optimal UE n^* for pattern j. We allocate pattern j to UE n^*, and set $s_{n^*,j}^* = 1$. Thus, the optimal solution $\mathbf{s}_j^* = \{s_{n^*,j}^*\}$ to the jth sub-problem in $(\mathcal{P}_\mathbf{8})$ is given by

$$s_{n^*,j}^* = \begin{cases} 1, & n^* = \arg\min_n \{\Gamma_{n,j}\} \\ 0, & otherwise \end{cases}. \tag{35}$$

Performing the procedure (29)-(35) for every pattern $j \in \mathcal{J}$, we obtain $\mathbf{S}^* = \{\mathbf{s}_1^*, ...\mathbf{s}_J^*\}$.

B. Heuristic Algorithm to Extract the Optimal Pattern (HAEOP)

Note that in $\mathbf{S}^*$, one UE may be allocated with more than one pattern, while in the LTE-A uplink one UE can be allocated with at most one pattern (as in (C7)), and the patterns allocated to different UEs should not contain the same RBs (as in (C8)), otherwise conflict will occur. A conflict table of each RB in 4-RB case is listed below in Table II.

TABLE II: Conflict table in 4-RB case

Index of RB	Corresponding conflicting patterns
1	2,6,9,11
2	3,6,7,9,10,11
3	4,7,8,9,10,11
4	5,8,10,11

We propose a heuristic algorithm called **h**euristic **a**lgorithm to **e**xtract the **o**ptimal **p**attern (HAEOP) for each UE to pick out the optimal pattern from their feasible patterns subjecting to constraints (C7) and (C8). It is given in Algorithm 3 and explained below.

In the sorting process (Step 4), we will give a higher priority to the UE with less feasible patterns, since UEs with more feasible patterns have a higher probability of finding a feasible pattern after all other UEs have performed their pattern selection.

According to $\mathbf{W}^n$, a pattern with a smaller index contains less RBs. So in Steps 5-6, the first $1 \sim (N_2 - 1)$ UEs will choose the pattern with the smallest index among their feasible patterns, so that the remaining UEs may have more chance of

Algorithm 3 HAEOP

```
1: Input: The obtained $\mathbf{S}^*$.
2: List the conflict table for each RB according to $\mathbf{W}^n$.
3: List the initial feasible pattern set for each remote-processing UE according to $\mathbf{S}^*$.
4: Sort the $N_2$ remote-processing UEs according to the number of their initial feasible patterns. The less it is, the top the UE. If multiple UEs have the same number of feasible patterns, compare the minimum feasible pattern index. The smaller it is, the top the UE.
5: Start the first round of pattern selection among all the valid remote UEs. If a UE has no feasible pattern, it is deemed as an invalid UE, and allocate pattern $j = 1$ to it. If there's only one remote UE, choose pattern $j = J$ as its final scheme, then break. Else, the first UE chooses the minimum pattern index from all its feasible patterns.
6: For the middle $2 \sim (N_2 - 1)$ UEs, perform pattern selection according to the following rules. (i) First each UE obtains all patterns conflicting with any its previous UEs; (ii) take out all the conflict patterns from its initial feasible pattern set, and the rest constitutes its new feasible pattern set; (iii) chooses the pattern with the minimum index from its new feasible pattern set; if the set is empty, allocate pattern $j = 1$ to the UE.
7: The last UE first performs the same procedure as the previous $2 \sim (N_2 - 1)$ UEs did to obtain its feasible pattern set. If the set is nonempty, choose the maximum index from this set; else choose pattern $j = 1$.
8: Calculate the total number of occupied RBs according to the chosen pattern of each UE. If it is less than $K$, start the next round of pattern selection: the first UE chooses its next feasible pattern; then the rest valid UEs repeat the same procedure as Steps 6 and 7 above, until either of the following two terminal conditions are satisfied: (i) the total number of occupied RBs equals to $K$, then the patterns selected in this round are considered as the optimal pattern allocation scheme; or (ii) if any valid UE has no feasible pattern, then the patterns selected in the previous round are considered as the optimal scheme.
9: Output: The optimal pattern $\mathbf{S}^{**} = \{s_{n^*,j^*}^{**}\}$.
```

finding a feasible pattern. In Step 7, the last UE will choose the feasible pattern with the maximum index in order to maximize RB utilization. If any UE is left with no feasible pattern in the first round of selection, i.e., all its initial feasible patterns conflict with the chosen patterns of its previous UEs, then the UE is allocated with pattern $j = 1$ (no RBs) and is called an invalid UE (i.e., it fails in task offloading), otherwise it is called a valid UE.

Step 8 terminates under one of the two conditions: 1) someone is left with no feasible pattern, then the patterns of all remote-processing UEs selected in the previous round are considered as the optimal pattern allocation scheme; 2) all the K RBs have been allocated, then the patterns selected in this round are considered as the optimal scheme. We denote the optimal pattern allocation matrix as $\mathbf{S}^{**} = \{s^{**}_{n^*,j^*}\}$.

After the optimal RB pattern allocation $\mathbf{S}^{**}$ is obtained, we perform the optimal power allocation for each UE n^* on the RBs in its selected pattern j^* as follows

$$p^{**}_{n^*,k(j^*)} = \begin{cases} p^*_{n,k(j)}, & n = n^* \ and \ j = j^* \\ 0, & otherwise \end{cases} . \quad (36)$$

C. Lagrange Multipliers Update

After solving all subproblems in $(\mathcal{P}_{\mathbf{8}})$, $\mathbf{S}$ and $\mathbf{P}$ can be obtained for given $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$. The dual variables $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ can be updated by resolving the dual problem of $(\mathcal{P}_{\mathbf{7}})$, which is given by

$$\begin{aligned} (\mathcal{P}_{\mathbf{10}}): \ & \max_{\boldsymbol{\mu},\boldsymbol{\omega}} \ D(\boldsymbol{\mu},\boldsymbol{\omega}) \\ \text{s.t.} \ & \boldsymbol{\mu} \succeq 0, \boldsymbol{\omega} \succeq 0. \end{aligned} \quad (37)$$

From (27) and (28), we know that $(\mathcal{P}_{\mathbf{10}})$ is convex, because $D(\boldsymbol{\mu},\boldsymbol{\omega})$ is a linear function about the dual variables $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$. By utilizing sub-gradient projection method, we solve $(\mathcal{P}_{\mathbf{10}})$ in an iterative manner to obtain dual optimum $\boldsymbol{\mu}^*$ and $\boldsymbol{\omega}^*$.

Proposition 3: The sub-gradients of $D(\boldsymbol{\mu},\boldsymbol{\omega})$ at the tth iteration are given in equations (38) and (39), where $p^*_{n,k(j)}$ and $s^*_{n,j}$ is the optimal solution to dual function (28) for a given set of dual variables $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$.

Proof: See Appendix C. ■

Based on (38)–(39), the Lagrange multipliers are updated with the sub-gradient projection method [30] as follows

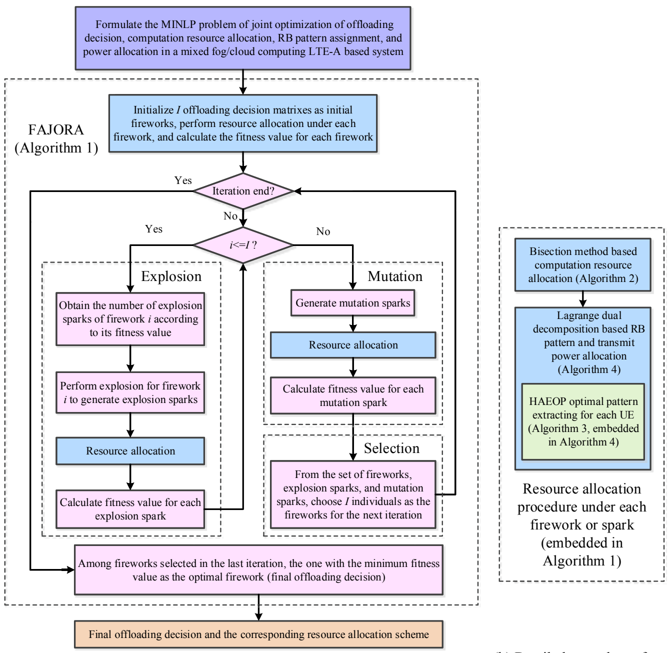
$$\mu_n(t+1) = \left[\mu_n(t) - h(t) \bigtriangledown \mu_n(t)\right]^+, \ \forall n, \quad (40)$$

$$\omega_n(t+1) = \left[\omega_n(t) - j(t) \bigtriangledown \omega_n(t)\right]^+, \ \forall n, \quad (41)$$

where t is the iteration index; $h(t)$ and $j(t)$ are positive step sizes. In this paper we adopt square summable but not summable step sizes [30], where $h(t) = 1/(10 * t)$, and $j(t) = 1/(10^{-1} * t)$. The Lagrange multipliers are updated iteratively until the required precision is satisfied. The procedure for joint RB pattern and power allocation is summarized in Algorithm 4. For a general understanding, the work flow chart of our system is given in Fig. 2, the main body of which is our proposed FAJORA.

Algorithm 4 Joint Uplink RB Pattern assignment and Power Allocation

```
Initialization:
 1: Set $\boldsymbol{\mu}(0), \boldsymbol{\omega}(0)$ and the precision $\delta$, set $t = 0$.
Iteration:
 2: while 1 do
 3:   for each pattern $j = 1$ to $\frac{K^2}{2} + \frac{K}{2} + 1$ do
 4:     for each $n$ do
 5:       if $k \in j$ then
 6:         Calculate $p^*_{n,k(j)}$ via (33) and obtain $\mathbf{\Gamma}_j$ by (34).
 7:         Obtain $s^*_{n^*,j}$ according to (35), and $\mathbf{s}^*_j = \{s^*_{n^*,j}\}$.
 8:         Extract $\mathbf{S}^{**} = \{s^{**}_{n^*,j^*}\}$ from $\mathbf{S}^* = \{\mathbf{s}^*_1, ..., \mathbf{s}^*_J\}$
            using Algorithm 3.
 9:         For $n = n^*$ and $j = j^*$, set $p^{**}_{n^*,k(j^*)} = p^*_{n,k(j)}$
            and $s^{**}_{n^*,j^*} = 1$.
10:       end if
11:     end for
12:   end for
13:   Update dual variables $\boldsymbol{\mu}, \boldsymbol{\omega}$ from (40) and (41), respectively.
14:   $t = t + 1$.
15:   if $\|\boldsymbol{\mu}(t+1) - \boldsymbol{\mu}(t)\|_2 < \delta$, $\|\boldsymbol{\omega}(t+1) - \boldsymbol{\omega}(t)\|_2 < \delta$
      then
16:     break.
17:   end if
18: end while
19: Output: $\mathbf{S}^{**} = \{s^{**}_{n^*,j^*}\}$, $\mathbf{P}^{**} = \{p^{**}_{n^*,k(j^*)}\}$.
```


Formulate the MINLP problem of joint optimization of offloading decision, computation resource allocation, RB pattern assignment, and power allocation in a mixed fog/cloud computing LTE-A based system
FAJORA (Algorithm 1)
Initialize I offloading decision matrixes as initial fireworks, perform resource allocation under each firework, and calculate the fitness value for each firework
Iteration end?
Yes
No
i<=I ?
Yes
No
Explosion
Obtain the number of explosion sparks of firework i according to its fitness value
Perform explosion for firework i to generate explosion sparks
Resource allocation
Calculate fitness value for each explosion spark
Mutation
Generate mutation sparks
Resource allocation
Calculate fitness value for each mutation spark
Selection
From the set of fireworks, explosion sparks, and mutation sparks, choose I individuals as the fireworks for the next iteration
Among fireworks selected in the last iteration, the one with the minimum fitness value as the optimal firework (final offloading decision)
Final offloading decision and the corresponding resource allocation scheme
Bisection method based computation resource allocation (Algorithm 2)
Lagrange dual decomposition based RB pattern and transmit power allocation (Algorithm 4)
HAEOP optimal pattern extracting for each UE (Algorithm 3, embedded in Algorithm 4)
Resource allocation procedure under each firework or spark (embedded in Algorithm 1)

(a) Main framework of FAJORA (b) Detailed procedure of resource allocation

Fig. 2: The work flow chart of our system.

VII. Complexity Analysis

The computational complexity of FAJORA in Algorithm 1 mainly comes from the resource allocation procedures in Steps 9 and 15 in the 'while' loop. In Step 9, the resource allocation

$$\bigtriangledown\mu_n(t) = p_n^{max} - \sum_{j=1}^{J} s_{n,j}^* p_{n,j}^*, \tag{38}$$

$$\bigtriangledown\omega_n(t) = \sum_{j=1}^{J} s_{n,j}^* r_{n,j} - \frac{D_n(y_n+z_n)}{\tau_1 - V_n} = \sum_{j=1}^{J}\sum_{k=1}^{K} s_{n,j}^* W_0 \log_2\left(1+p_{n,k(j)}^* g_{n,k(j)}\right) - \frac{D_n(y_n+z_n)}{\tau_1 - V_n}, \tag{39}$$

procedure is performed $\sum_{i=1}^{I} i\chi_i$ times for the $\sum_{i=1}^{I} i\chi_i$ explosion sparks, respectively. In Step 15, the resource allocation procedure is performed γ times for the γ mutation sparks, respectively. For notational simplicity, we define the total number of explosion and mutation sparks as $\Xi = \sum_{i=1}^{I} i\chi_i + \gamma$.

In each resource allocation procedure, computation resources are allocated using Algorithm 2, and then radio resources are allocated employing Algorithm 4. In Algorithm 2, it requires $O\left(\log_2\left(\frac{\tau^{max}-\tau^{min}}{\varepsilon}\right)\right)$ iterations for the bisection method to converge.

In Algorithm 4, the complexity mainly comes from the extracting of RB pattern in Step 8, i.e., Algorithm 3. The sub-gradient projection method in the outer 'while' loop that needs $O\left(\frac{1}{\delta^2}\right)$ iterations to converge [28], the $\frac{K^2}{2}+\frac{K}{2}+1$ iterations in the outer 'for' loop, and the at most N iterations in the inner 'for' loop. The complexity of Algorithm 3 mainly comes form the pattern extracting procedure in its Steps 5–8. Since there are at most $N_2 = N$ remote-processing UEs, the complexity of the first round of RB pattern extracting in Steps 5–7 is $O(N)$. Assuming that the first remote-processing UE posses C feasible patterns, since all the remote-processing UEs are sorted according to the ascending order of their number of feasible patterns, C is far less than N, and thus the complexity of RB pattern extracting in Steps 5–8 is $O(CN) = O(N)$. Consequently, the complexity of Algorithm 3 is $O(N)$. Hence, the complexity of Algorithm 4 is $O\left(\frac{1}{\delta^2} * (\frac{K^2}{2}+\frac{K}{2}+1) * N * N\right) = O\left(\frac{1}{\delta^2}K^2N^2\right)$. Therefore, the complexity of each resource allocation procedure is $O\left(\log_2\left(\frac{\tau^{max}-\tau^{min}}{\varepsilon}\right)\right) + O\left(\frac{1}{\delta^2}K^2N^2\right) = O\left(\frac{1}{\delta^2}K^2N^2\right)$.

Based on the above analysis and given that the outer 'while' loop in Algorithm 1 runs for L times, the complexity of FAJORA is $O\left(\frac{1}{\delta^2}\Xi L K^2 N^2\right)$.

VIII. Results and Discussions

In this section, simulation results are presented to evaluate the performance of the proposed algorithms. The following parameters remain unchanged through our simulations: $L = 20$, $K = 15$ [9], $W_0 = 180$ KHz [9], $p_n^{max} = 2$ W [22].

The following parameters are set as default unless otherwise specified: $N = 6$, $I = 2$, $M = 4$, $\gamma = 1$, $a = 0.2$, $b = 0.8$, $F^{fog} = 5*10^9$ cycles/s [18], $f_n^c = 20*10^9$ cycles/s [16], f_n^{loc} is uniformly distributed in $[50, 400]$ M cycles/s, and $R_n^{fc} = 15*10^6$ b/s [20]. For simplicity, the wireless channel gain $g_{n,k(j)} = \frac{h_{n,k(j)}}{\sigma^2}$ is assumed to take values in $[5, 14]$ randomly [29]. We adopt face recognition [12] as the default application, where $D_n = 0.42$ MB and $\lambda_n = 297.62$ cycles/bit [12].

Next, we verify the performance gain obtained by our proposed algorithms and the following schemes are compared.

- *The proposed scheme (FAJORA)*: The scheme obtains offloading decisions and resource allocation using FAJORA in Algorithm 1, where in each iteration computation resource is allocated using Algorithm 2, RB pattern and power are allocated using Algorithm 4, and each UE picks out the optimum RB pattern using Algorithm 3.
- **R***adio and* **c***omputation* **r***esource* **a***llocation optimization (RCRA)*: RB pattern and power are allocated using Algorithms 3 and 4, computation resource are allocated using Algorithm 2, and offloading decisions are obtained randomly.
- **R***adio* **r***esource* **a***llocation with HAEOP (RRA-H)*: RB pattern and power are allocated employing Algorithms 3 and 4, while offloading decisions and computation resource allocation are obtained randomly.
- **R***adio* **r***esource* **a***llocation and* **r***andom pattern extracting (RRA-R)*: RB pattern and power are allocated using Algorithm 4, and from the allocated patterns each UE selects one of which randomly. Offloading decisions and computation resource allocation are obtained randomly.
- *Local processing (Local)*: All UEs process their applications locally without optimization.

Moreover, in the following Figs. 8 and 9, another algorithm named **SDR** based **o**ffloading **d**ecision optimization and optimized **r**esource **a**llocation algorithm (SDR-ODRA) was considered as a benchmark to demonstrate the performance of our proposed BTFA based offloading decision making algorithm. In SDR-ODRA, the offloading decisions are obtained using the SDR based algorithm in [20], where in each iteration computation resource is allocated using Algorithm 2, RB pattern and power are allocated using Algorithm 4, and each UE picks out the optimum RB pattern using Algorithm 3.

*R*emark: SDR based offloading decision making algorithm was first novelly proposed by the authors in [20], and now has been widely used in many existing works. Similar to our previous work [23], the number of runs (i.e., randomization trails) [23] in SDR-ODRA is set as 6.

Five metrics are adopted, including: 1) three kinds of delay, i.e., the maximum, minimum and average delay of all UEs, which are denoted as T_{max} (i.e., objective value), T_{min}, and T_{av}, respectively; 2) the number of benefited UEs, where a benefited UE means the UE whose delay consumption is reduced compared with local processing; 3) the probability of failure in offloading, where a UE fails in offloading means it is allocated with pattern $j = 1$.

A. Convergence of Algorithms 1, 2 and 4

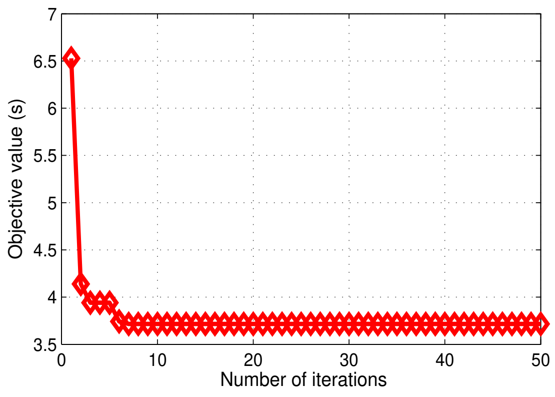


Fig. 3: Convergence of Algorithm 1 (FAJORA).

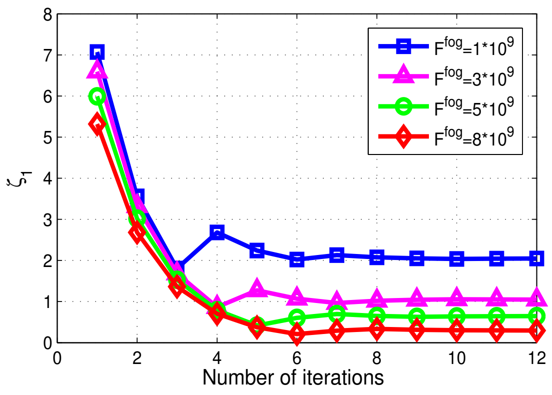


Fig. 4: Convergence of Algorithm 2.

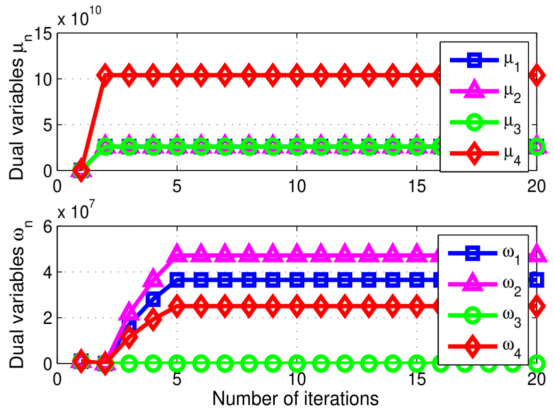


Fig. 5: Convergence of Algorithm 4.

Fig. 3 verifies the convergence of the outer loop of FAJORA in Algorithm 1, from which we can see FAJORA converges fast within 10 iterations. Figs. 4 and 5 evaluate the convergence rate of the main loop of Algorithms 2 and 4, respectively, both of which are embedded in Steps 3, 9 and 15 in Algorithm 1. As discussed in Section V, τ is the maximum delay of all fog-processing UEs, and Fig. 4 shows that τ decreases under different F^{fog} after each iteration until convergence. Fig. 5 shows that the dual variables in Algorithm 4 converge fast. According to the three figures, we know that the proposed algorithms are cost-efficient in solving the NP-hard problem $(\mathcal{P}_1)$.

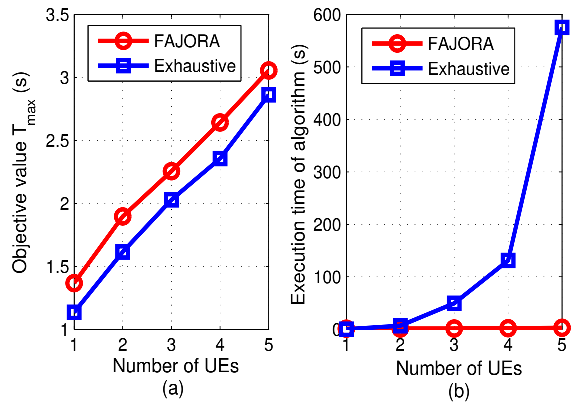


Fig. 6: Effectiveness and complexity of FAJORA.

Fig. 6 shows the comparisons in effectiveness and complexity between the proposed algorithm FAJORA and exhaustive algorithm, where offloading decisions are obtained by exhaustive search, and computation and communication resource allocation employ our proposed Algorithms 2, 3, and 4. From the Fig. 6(a), it can be known that FAJORA is slightly inferior to exhaustive search in performance, i.e., a little increase in objective value. However, the complexity comparison in Fig. 6(b) indicates that the execution time of exhaustive algorithm increases exponentially with the number of UEs, while FAJORA only takes a little execution time even with more UEs, indicating that is good in scalability.

B. Effectiveness of Algorithm 3 (HAEOP)

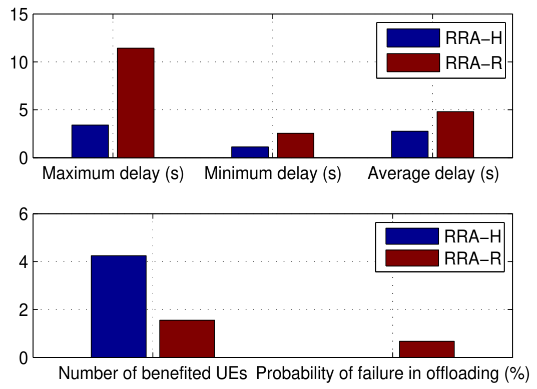


Fig. 7: Performance evaluation of Algorithm 3.

In Fig. 7 the performance of HAEOP in Algorithm 3 is evaluated by comparing RRA-H and RRA-R, where all the parameters adopt their default values. As shown in the first sub-figure, the three delays T_{max}, T_{min} and T_{av} of RRA-H are always much shorter than that of RRA-R. The second sub-figure indicates RRA-H could benefit more UEs and reduce the probability of failures in offloading effectively.

The reason for Fig. 7 is that: in RRA-R, each UE selects a random pattern from its feasible patterns, thus its chosen pattern may contain the same RB with the patterns chosen by its previous UEs, and consequently conflict will happen, leading to higher failure probability and less benefited UEs. While in RRA-H, since HAEOP is adopted in pattern selection, each UE picks out the optimum feasible pattern, considering exclusiveness of RBs, thus failures could be avoided effectively, and therefore more UEs will be benefited as is shown in the second sub-figure. On the other hand, HAEOP takes RB utilization into account, thus the maximum RB utilization can be obtained under the final selected pattern allocation scheme. Consequently, T_{max}, T_{min} and T_{av} can be reduced greatly as shown in the first sub-figure.

C. Performance Comparisons versus Different Application parameters

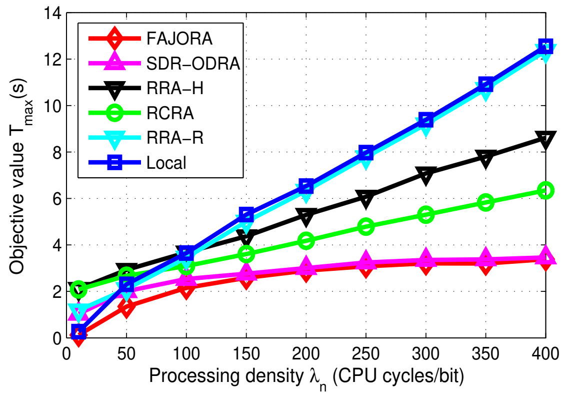


Fig. 8: Objective value T_{max} comparison under different processing density λ_n.

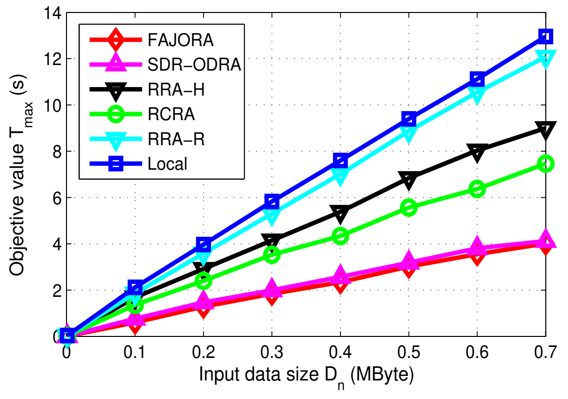


Fig. 9: Objective value T_{max} comparison under different input data size D_n.

Figs. 8 and 9 shows how application parameters including processing density λ_n and input data size D_n affect the objective value T_{max}, respectively. The two figures are in accordance with our intuition that more input data D_n or the higher computation complexity λ_n, higher delay will be brought in, and consequently a lager value of T_{max}. Moreover, as a joint optimization of offloading decisions and resource allocation, FAJORA performs always the best, followed by SDR-ODRA, RCRA, RRA-H, and RRA-R successively, and Local is the worst in performance.

However, some differences exist between the two figures. In Fig. 8, D_n takes the default value 0.42 MB, which is relatively large. When λ_n is very small, local processing is usually a good choice, while offloading will consume more time in data transmission. In Fig. 9, $\lambda_n = 297.62$ cycles/bit. When D_n is very small, the computation workload C_n is also very small, so all the algorithms will consumes quite less time, and therefore T_{max} is very small for all the algorithms.

It should be noted that, although SDR-ODRA can obtain almost the same performance as FAJORA, the computational complexity of SDR-ODRA is much higher than FAJORA in the offloading decision making process. In SDR-ODRA, offloading decisions are obtained using CVX and randomization, where interior point method is adopted, leading to higher complexity. In FAJORA, the outer fireworks algorithm need several iterations to converge, and in each iteration each spark (i.e., an offloading decision) can be generated using fireworks operators with very tiny complexity.

D. Performance Comparisons versus Channel State

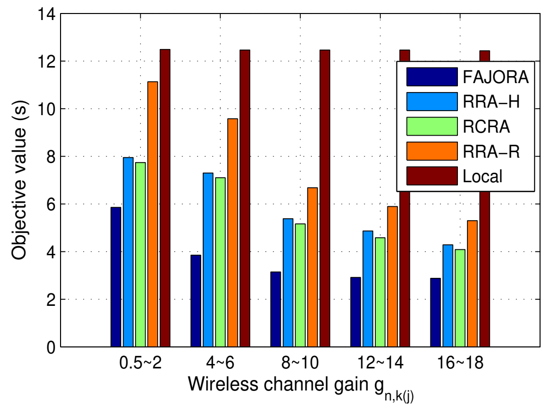


Fig. 10: Objective value T_{max} comparison under different different wireless channel gain $g_{n,k(j)}$.

Figs. 10 and 11 display how channel state affects the objective value T_{max}, including the wireless access channel gain $g_{n,k(j)}$ between UEs and the fog node, and the wired link rate R_n^{fc} between fog and cloud, respectively. As channel state has no influence on local processing, its objective value always keeps still. However, when the channel state gets better and better, less time will be consumed in data transmission for all other algorithms, leading to a decrease in T_{max} for them. From the two figures we can also find that FAJORA always performs the best, with its objective value T_{max} far less than other algorithms.

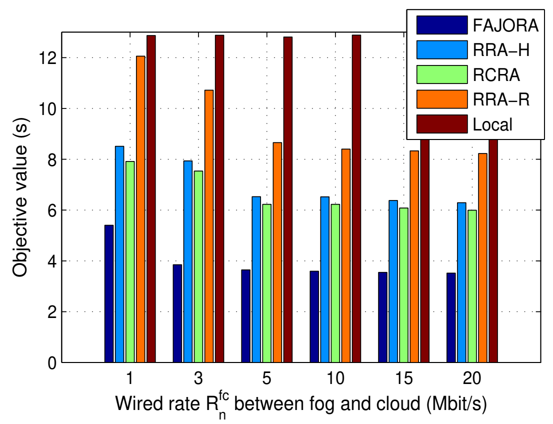


Fig. 11: Objective value T_{max} comparison under different different wired rate R_n^{fc} between fog and cloud.

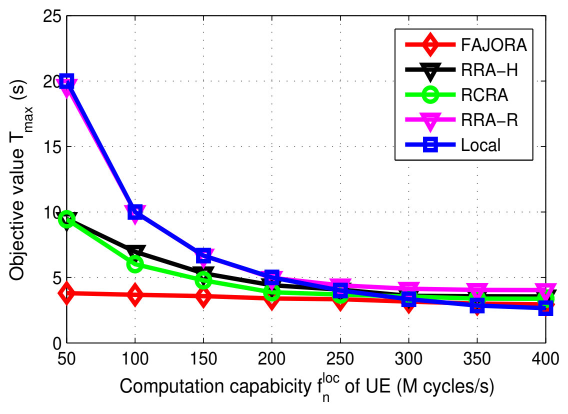


Fig. 12: Objective value T_{max} comparison under different local processing capability f_n^{loc}.

E. Performance Comparisons versus the Processing Capabilities

The impact of local processing capability f_n^{loc} on T_{max} is shown in Fig. 12, where T_{max} decreases quickly with the increase of f_n^{loc} for all the algorithms. When the value of f_n^{loc} is large, Local performs the best in delay reduction. This is reasonable, because when the processing capability of a UE is strong enough, it is capable in processing most applications with good performance, and there's no need to offload. However, the obtained T_{max} of FAJORA is only a little higher than Local, indicating FAJORA performs well in this case. On the other hand, when f_n^{loc} is very small, FAJORA still performs very well, whereas the delay of all other algorithms are too long to bear.

In Figs. 13 and 14, we evaluate the impact of the fog processing capability F^{fog} and cloud processing ability f_n^c on T_{max}. When either of the two parameters increase, T_{max} decrease, which is the same for all the algorithms (except for Local) and is in line with our intuition. Besides, FAJORA performs the best in delay reduction and far outdistances other algorithms.

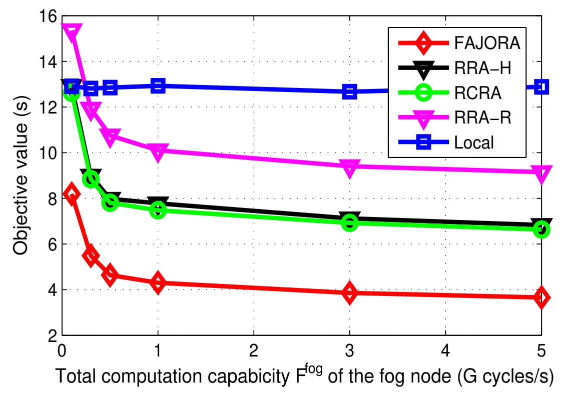


Fig. 13: Objective value T_{max} comparison under different total fog processing capability F^{fog}.

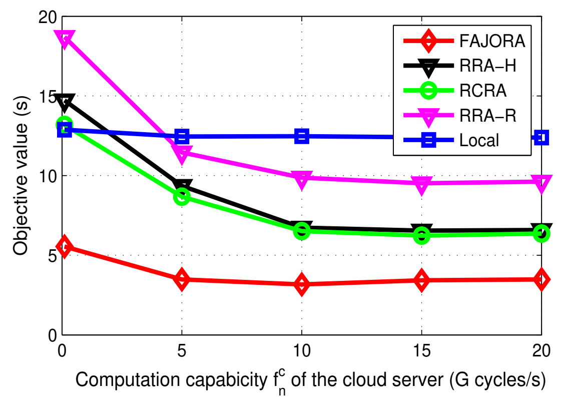


Fig. 14: Objective value T_{max} comparison under different cloud processing capability f_n^c.

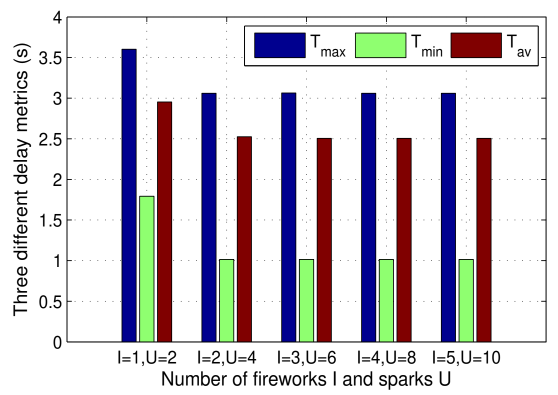


Fig. 15: Objective value T_{max} comparison under different number of fireworks I and sparks M.

F. Performance Comparisons versus FA's Parameters

Fig. 15 presents the influence of the parameters in fireworks algorithm on three different delay metrics T_{max}, T_{min} and T_{av}. For notational simplicity, we define the total number of explosion and mutation sparks as $U = \sum_{i=1}^{I} i\chi_i + \gamma$. As is shown, with the number of fireworks I and sparks U increase, the three delays all decrease. This is because, as was mentioned, each of the fireworks or sparks is an offloading decision matrix and corresponds to a resource allocation scheme. So the more fireworks and sparks, the more joint computation offloading and resource allocation schemes, consequently the better searching capabilities and the shorter obtained delays. However, the more fireworks and sparks, the more computation complexity, whereas the delay reduction is not so significantly as is shown in Fig. 15. Thus we choose $I = 2$ and $U = 4$ as the default number of fireworks and sparks, respectively, to strike a balance between searching capability and computation complexity.

IX. Conclusions

In this paper, we have proposed a framework to optimize computation offloading, computation resource allocation, RB pattern assignment, and transmit power allocation. In the optimization framework, we have considered the maximum delay reduction problem, which was modeled as an MINLP problem. We have proposed a low-complexity general algorithm framework FAJORA to decompose it into several subproblems, where offloading decisions was obtained within the main framework of FAJORA, and computation and radio resources allocation was solved by the embedded Algorithms 2, 3, and 4. Abundant simulation results have demonstrated the convergence and effectiveness of our proposed algorithms.

Appendix A
Proof of Proposition 1

Proof: In problem $(\mathcal{P}_{\mathbf{1}})$, there are four sets of mutually coupled variables to be optimized. If the optimal offloading decision $\mathbf{\Pi}^*$ and computation resource allocation $\mathbf{f}^{fog^*}$ are given, then $(\mathcal{P}_{\mathbf{1}})$ reduces to the joint optimization of RB pattern assignment and transmit power control among all the remote-processing UEs in $\mathcal{N}_2$. If the optimal RB pattern assignment $\mathbf{S}^*$ is also obtained, then the problem can be further reduced to

$$\begin{aligned}(\mathcal{P}_{\mathbf{1}}^{\mathbf{1}}):\ &\min_{\mathbf{P}} \max_{n\in\mathcal{N}_2} \left(\frac{D_n}{r_n} + V_n\right) \qquad (42)\\ \text{s.t.}\ \ &(\text{C9}): p_{n,k(j)} \geq 0, \forall n \in \mathcal{N}_2, \forall k \in \mathcal{K}, \forall j \in \mathcal{J},\\ &(\text{C10}): \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} p_{n,k(j)} \leq p_n^{max}, \forall n \in \mathcal{N}_2,\end{aligned}$$

where $V_n = \frac{C_n y_n}{f_n^{fog}} + \frac{C_n x_n}{f_n^{loc}} + (T_n^{fc} + T_n^{c}) z_n = \frac{C_n y_n}{f_n^{fog}} + (T_n^{fc} + T_n^{c}) z_n$ is a constant, so problem $(\mathcal{P}_{\mathbf{1}}^{\mathbf{1}})$ is equivalent to

$$\begin{aligned}(\mathcal{P}_{\mathbf{1}}^{\mathbf{2}}):\ &\min_{\mathbf{P}} \max_{n\in\mathcal{N}_2} \frac{D_n}{r_n} \qquad (43)\\ \text{s.t.}\ \ &(\text{C9}), (\text{C10}).\end{aligned}$$

In the following, we show that problem $(\mathcal{P}_{\mathbf{1}}^{\mathbf{2}})$ is NP-hard. As a special case, we assume there is no $\max\limits_{n\in\mathcal{N}_2}$ operation, then problem $(\mathcal{P}_{\mathbf{1}}^{\mathbf{2}})$ becomes

$$\begin{aligned}(\mathcal{P}_{\mathbf{1}}^{\mathbf{3}}):\ &\min_{\mathbf{P}} \frac{D_n}{\sum\limits_{k\in\mathcal{K}} W_0 \log_2\left(1 + \frac{p_{n,k(j)} h_{n,k(j)}}{\sigma^2}\right)} \qquad (44)\\ \text{s.t.}\ \ &(\text{C9}), (\text{C10}),\end{aligned}$$

which can be transformed into

$$\begin{aligned}(\mathcal{P}_{\mathbf{1}}^{\mathbf{4}}):\ &\max_{\mathbf{P}} \frac{1}{D_n} \sum_{k\in\mathcal{K}} W_0 \log_2\left(1 + \frac{p_{n,k(j)} h_{n,k(j)}}{\sigma^2}\right) \qquad (45)\\ \text{s.t.}\ \ &(\text{C9}), (\text{C10}).\end{aligned}$$

We can see that the objective function in $(\mathcal{P}_{\mathbf{1}}^{\mathbf{4}})$ is a sigmoidal function.

Definition: A continuous function $f[l, u] \to \mathcal{R}$ is defined as a sigmoidal if: either it is convex, concave, or convex for $x \leq z, z \in [l, u]$ and concave for $x \geq z$ [31], [32].

Since all the constraints of $(\mathcal{P}_{\mathbf{1}}^{\mathbf{4}})$ are linear, $(\mathcal{P}_{\mathbf{1}}^{\mathbf{4}})$ maximizes the sum of a set of sigmoidal functions over a convex set, which is a sigmoidal programming problem and is NP-hard [31], [32]. Consequently, problem $(\mathcal{P}_{\mathbf{1}})$ is NP-hard, and Proposition 1 holds.

■

Appendix B
Proof of Proposition 2

Proof: When $f(x)$ is concave, then the perspective function $g(x, t) = tf(x/t)$ is concave, too [28]. Since $s_{n,j} \log_2(1 + \frac{\phi_{n,k(j)} g_{n,k(j)}}{s_{n,j}})$ is the perspective function of the concave function $\log_2(1 + \phi_{n,k(j)} g_{n,k(j)})$, it preserves concavity, too. As the sum of several concave functions is still concave, $\sum_{j=1}^{J}\sum_{k=1}^{K} s_{n,j} \log_2(1 + \frac{\phi_{n,k(j)} g_{n,k(j)}}{s_{n,j}})$ is also concave. On the other hand, the super-level set of concave function is convex [28], so (C18) is convex. Moreover, (C5) − (C10) are all linear constraints. Thus, $(\mathcal{P}_{\mathbf{7}})$ is a convex optimization programming that minimize a convex function over a convex set. ■

Appendix C
Proof of Proposition 3

Proof: Observing the definition of $D(\boldsymbol{\mu}, \boldsymbol{\omega})$ of (28), we have

$$D(\boldsymbol{\mu}', \boldsymbol{\omega}') \geq \tau_1 + \sum_{n\in\mathcal{N}_2} \mu_n' \left(p_n^{max} - \sum_{j=1}^{J}\sum_{k=1}^{K} \phi_{n,k(j)}^*\right) + \sum_{n\in\mathcal{N}_2} \omega_n' \left[\sum_{j=1}^{J}\sum_{k=1}^{K} W_0 s_{n,j}^* \log_2\left(1 + \frac{\phi_{n,k(j)}^* g_{n,k(j)}}{s_{n,j}^*}\right) - \frac{D_n(y_n + z_n)}{\tau_1 - V_n}\right]. \quad (46)$$

Rearranging (46), we have

$$D(\boldsymbol{\mu}',\boldsymbol{\omega}')\geq D(\boldsymbol{\mu},\boldsymbol{\omega})+\sum_{n\in\mathcal{N}_2}(\mu_n'-\mu_n)\left(p_n^{max}-\sum_{j=1}^{J}\sum_{k=1}^{K}\phi_{n,k(j)}^*\right)$$
$$+\sum_{n\in\mathcal{N}_2}(\omega_n'-\omega_n)\left[\sum_{j=1}^{J}\sum_{k=1}^{K}W_0 s_{n,j}^*\log_2\left(1+\frac{\phi_{n,k(j)}^* g_{n,k(j)}}{s_{n,j}^*}\right)\right.$$
$$\left.-\frac{D_n(y_n+z_n)}{\tau_1-V_n}\right]. \quad (47)$$

Note that a sub-gradient η of a convex function $f(\cdot)$ is defined as: if $f(x)\geq f(y)+\eta^T(x-y),\forall x,y$. Thus, Proposition 3 holds. ■

References

[1] J. Kwak, Y. Kim, J. Lee, et al., "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," *IEEE J. on Sel. Areas in Commun.*, vol. 33, no. 12, pp. 2510-2523, 2015.

[2] H. T. Dinh , C. Lee, D. Niyato et al., "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587-1611, 2013.

[3] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," *Springer Internet of Everything*, 103-130, 2018.

[4] W. Shi, J. Cao, Q. Zhang, et al., "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.

[5] T. X. Tran, A. Hajisami, P. Pandey, et al., "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Commun. Mag.*, vol. 55, no, 4, pp. 54-61, 2017.

[6] B. G. Chun, S. Ihm, P. Maniatis, et al., "Clonecloud: Elastic Execution Between Mobile Device and Cloud," in *ACM Proc. 6th Eur. Conf. Comput. Syst.*, pp. 301-314, 2011.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE TPervasive Comput.*, vol. 8, no. 4, pp. 14-23, 2009.

[8] F. Ghavimi, Y. W. Lu, and H. H. Chen, "Uplink Scheduling and Power Allocation for M2M Communications in SC-FDMA based LTE-A Networks with QoS Guarantees," *IEEE Trans. on Veh. Technol.*, vol. 66, no. 7, pp. 6160-6170, 2017.

[9] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*, 3GPP TS 36.211 V8.6.0 Std., Mar. 2009.

[10] E. Cuervo, A. Balasubramanian, D. Cho, et al., "MAUI: Making Smartphones Last Longer with Code Offload," in *ACM Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, pp. 49-62, 2010.

[11] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81-93, 2015.

[12] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974-983, 2015.

[13] K. Liu, X. Zhang, and Z. Huang. "A Combinatorial Optimization for Energy-Efficient Mobile Cloud Offloading over Cellular Networks," in *Proc. IEEE GLOBECOM*, pp. 1-6, 2016.

[14] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Trans. Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89-103, 2015.

[15] S. Guo, B. Xiao, Y. Yang, et al., "Energy-Efficient Dynamic Offloading and Resource Scheduling in Mobile Cloud Computing," in *Proc. IEEE INFORCOM*, pp. 1-9, 2016.

[16] X. Lyu, H. Tian, C. Sengul, et al., "Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds," *IEEE Trans. on Veh. Technol.*, vol. 66, no. 4, pp. 3435-3447, 2017.

[17] C. Wang, F. R. Yu, C. Liang, et al., "Joint Computation Offloading and Interference Management in Wireless Cellular Networks With Mobile Edge Computing," *IEEE Trans. on Veh. Technol.*, 2017.

[18] P. Zhao, H. Tian, C. Qin, et al., "Energy-Saving Offloading by Jointly Allocating Radio and Computational Resources for Mobile Edge Computing," *IEEE Access*, 2017.

[19] J. Cheng, Y. Shi, B. Bai, et al, "Computation Offloading in Cloud-RAN Based Mobile Cloud Computing System," in *Proc. IEEE ICC*, pp. 1-6, 2016.

[20] M. H. Chen, B. Liang, and M. Dong, "Joint Offloading Decision and Resource Allocation for Multi-User Multi-Task Mobile Cloud," in *Proc. IEEE ICC*, pp. 1-6, 2016.

[21] Y. Li, M. Sheng, C. W. Tan, et al., "Energy-Efficient Subcarrier Assignment and Power Allocation in OFDMA Systems with Max-min Fairness Guarantees," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3183-3195, 2015.

[22] Y. Mao, J. Zhang, and K. B. Letaief. "Dynamic Computation Offloading For Mobile-Edge Computing with Energy Harvesting Devices," *IEEE J. on Sel. Areas in Commun.*, vol. 34, no. 12, pp. 3590-3605, 2016.

[23] J. Du, L. Zhao, J. Feng and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee." *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594 - 1608, 2018.

[24] Y. Tan, and Y. Zhu, "Fireworks Algorithm for Optimization," in *Advances in Swarm Intelligence, Springer*, pp. 355-364, 2010.

[25] N. Sharma, and A.S. Madhukumar, "Genetic Algorithm Aided Proportional Fair Resource Allocation in Multicast OFDM Systems," *IEEE Trans. Broadcasting*, vol. 61, no. 1, pp. 16-29, 2015.

[26] Y.J. Gong, J. Zhang, H. S. H. Chung, et al., "An Efficient Resource Allocation Scheme Using Particle Swarm Optimization," *IEEE Trans. Evolutionary Computation*, vol. 16, no. 6, pp. 801-816, 2012.

[27] S. C. Wang, and Y. H. Liu, "A PSO-Based Fuzzy-Controlled Searching for The Optimal Charge Pattern of Li-ion Batteries," *IEEE Trans. Industrial Electronics*, vol. 62, no. 5, pp. 2983-2993, 2015.

[28] S. Boyd and L. Vandenberghe, "Convex Optimization," Cambridge, U.K.: Cambridge Univ. Press, 2004.

[29] D.S. Zhai, M. Sheng, X Wang, et al., "Leakage-Aware Dynamic Resource Allocation in Hybrid Energy Powered Cellular Networks," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4591-4603, 2015.

[30] S. Boyd, "Subgradient Methods," [Online]. Lecture notes of EE364b, Stanford University, Winter Quarter 2007 (2006).

[31] M. Udell, and S. Boyd, S, "Maximizing a sum of sigmoids," *Optimization and Engineering*, 2013.

[32] Y. Li, M. Sheng M, X. Wang, et al., "Max-min energy-efficient power allocation in interference-limited wireless networks", *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 4321-4326, 2015.