Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G

Lichao Yang, Heli Zhang, *Member, IEEE*, Ming Li*, Member, IEEE*, Jun Guo, and Hong Ji*, Senior Member, IEEE*

***Abstract*—Mobile edge computing has been proposed in recent years to offload computation tasks from user equipments (UEs) to the network edge to break hardware limitations and resource constraints at UEs. Although there have been some existing works on computation offloading in 5G, most of them fail to take into account the unique property of 5G in their scheme design. In this paper, we consider small-cell network architecture for task offloading. In order to achieve energy efficiency, we model the energy consumption of offloading from both task computation and communication aspects. Besides, transmission scheduling are carried over both the fronthaul and backhaul links. We first formulate an energy optimization problem of offloading, which aims at minimizing the overall energy consumption at all system entities and takes into account of the constraints from both computation capabilities and service delay requirement. We then develop an artificial fish swarm algorithm based scheme to solve the energy optimization problem. Besides, the global convergence property of the our scheme is formally proven. Finally, various simulation results demonstrate the efficiency of our scheme.**

***Index Terms*—Mobile edge computing (MEC), small-cell network, fronthaul and backhaul links, AFSA based scheme.**

I. Introduction

With the striking rise in penetration of user equipments (UEs), there exist tons of mobile applications nowadays, such as natural language processing, virtual reality and interactive gaming [1]–[3]. On the one hand, they provide users with diverse services and experiences. On the other hand, the limited battery power, computation capacity and cache size at UEs largely hinder the wide usage of these applications, especially the ones which are computation-intensive. Meanwhile, mobile edge computing (MEC) emerges as a new network architecture concept that provides UEs with cloud computing capabilities and an Information technology (IT) service environment at the edge of the cellular network. By offloading computation tasks from UEs to MEC, it can alleviate resource constraints at UEs and better facilitate them to enjoy various applications. Besides, the 5th generation wireless systems, abbreviated 5G, are the proposed next telecommunications standards beyond the current 4G/IMT-Advanced standards. Therefore, it is desirable to study the integration of MEC and 5G to support the ever-increasing amount of mobile applications.

There have been quite a few existing researches on computation offloading for MEC in 5G communication systems [4]–[9]. However, most of them fail to take into account of unique characters of 5G in their scheme design. Notice that the small cell serves as a key technology in future 5G. In particular, the small-cell network (SCN) is composed of several tiers of heterogeneous small base stations (BSs), such as micro base stations (MBSs) and femto relay base stations (FRSs). Their roles in an SCN are not the same due to their different locations deployed and hardware equipped. In this research we aim to study computation offloading in 5G MEC, considering the property of SCN architecture. There are also limited works on MEC in SCN [10], [11]. However, none of them discusses energy efficiency or take into account of latency requirement from tasks.

As shown in Fig. 1, an SCN is typically a two-tier architecture composed of FRSs and MBSs. A UE relays its data through FRSs and MBSs to the core network. Thus, there are two segments of wireless links in a communication connection from a UE to the core network. Specifically, we call the one from an UE to an FRS as a fronthaul link and the one from an FRS to an MBS as a backhaul link. Besides, since MBSs are rich in computation and storage resources, they are ideal places to install with MEC servers that provide computation services for UEs. Under this architecture, to offload computation tasks from UEs to MEC servers, they travel through fronthaul links and backhaul links and arrive at MBSs. Therefore, it is critical to jointly select appropriate fronthaul and backhaul links for the computation offloading in order to achieve overall efficient resource allocation.

In this paper, we study the computation offloading in 5G MEC, featured by its SCN architecture. Specifically, we aim at achieving energy-efficient task offloading for all system entities. Thus, an energy cost minimization problem is formulated, taking into account of resource constraints at MEC servers and ser-

Manuscript received July 18, 2017; revised October 21, 2017 and December 15, 2017; accepted January 9, 2018. Date of publication January 30, 2018; date of current version July 16, 2018. This work was supported in part by the National Natural Science Foundation of China for the Youth under Grant 61501047, in part by the National Natural Science Foundation of China under Grant 61771070, and the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2016ZX03001017. The review of this paper was coordinated by Dr. P. Lin. *(Corresponding author: Heli Zhang.)*

L. Yang, H. Zhang, J. Guo, and H. Ji are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lichaoyang@bupt.edu.cn; zhangheli@bupt.edu.cn; junguo@bupt.edu.cn; jihong@bupt.edu.cn).

M. Li is with the Department of Computer Science and Engineering, University of Nevada, Reno 89557 USA (e-mail: mingli@unr.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2018.2799620

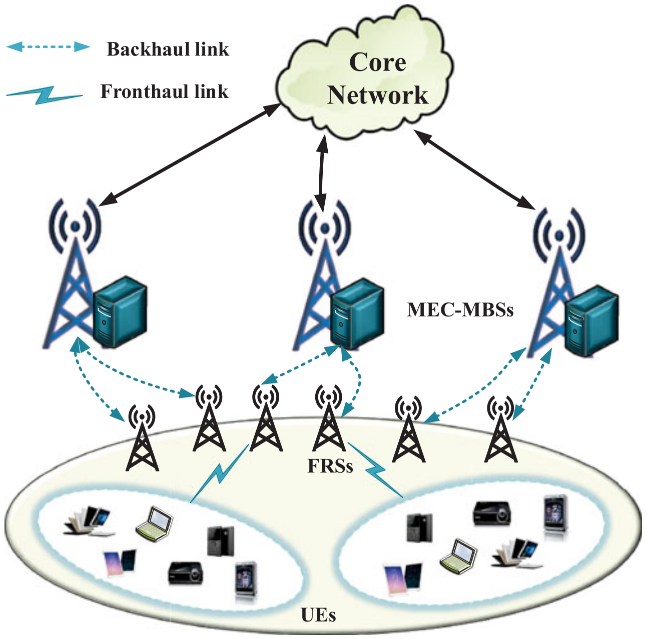

Fig. 1. Small-cell network scenario integrated with mobile edge computing.

vice latency requirements. Noticing the formulated optimization problem is a binary non-linear integer programming problem, which is NP-hard to solve, we then develop a heuristic algorithm based on the artificial fish swarm algorithm (AFSA). More importantly, we further prove that our algorithm achieves global convergence. Finally, simulations are conducted to demonstrate the performance of our proposed algorithm.

We summarize the main contributions of this paper as follows:

- We study energy-efficient offloading in 5G MEC, under its unique network architecture. The fronthaul and backhaul links are jointly considered when designing the computation offloading scheme.
- We formulate the energy optimization problem for offloading with the objective to minimize network overall energy cost and guarantee UE task's latency requirements. The energy and latency are related to the task transmission through fronthaul and backhaul, and also the task computation at the MEC server.
- We develop a heuristic algorithm to solve the energy optimization problem based on the artificial fish swarm algorithm. In addition, we formally prove the global convergence property of our proposed algorithm.

The rest of this paper is organized as follows. The related work is discussed in Section II. We then describe the system model in Section III. In Section IV, we formulate the energy optimization problem for 5G MEC and develop a heuristic algorithm based on the artificial fish swarm algorithm. We formally prove the convergence property of the proposed algorithm in Section V. Simulation results are discussed in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

There have been some existing works on computation offloading in MEC. For instance, in [12], the authors proposed a computation offloading scheme to determine whether the computation-intensive tasks should be performed locally or remotely on the MEC servers. The overall network energy consumption is minimized. In [13], the authors studied the dynamic service migration problem and used Markov decision process to decrease the latency of the service migration. In [14], the authors combined energy and latency together as the overhead and proposed a game-based algorithm to reducing the overall network overhead. In [15], an iterative algorithm was proposed to improve the resource utilization efficiency when offloading tasks to the MEC servers. In [14], [16], the multi-user computation offloading problem in the wireless environment with multiple channels was investigated. Authors of [17] proposed a method to jointly optimize the transmit power, the number of bits per symbol and the CPU cycles for each UE. Based on which, the power consumption at the UE side was lowered. The work [18] optimized the offloading decisions of all UEs as well as the allocation of communication resource. Meanwhile, the overall cost of energy, computation, and delay for all UEs were decreased. In [19], the computation and communication resource of UEs were shared with each other according to their social relationship. Note that the above schemes are not specifically designed for 5G communication systems.

Resource management in 5G MEC has also been studied recently. In [4], the authors considered improving the total network revenue by jointly utilizing the computation offloading, resource allocation and content caching together. The authors in [5] studied how to decrease the computation offloading latency in 5G MEC infrastructure. In [7], the MEC server is deployed at the edge of the cloud radio access network and a real-time, context-aware MEC server deployment mechanism is studied. Since the Internet of things (IoT) is recognized as a main driver for 5G, the integration of MEC and IoT was investigated [8], [9]. These works fail to take into account of SCN architecture into their scheme design.

There are only handful works on MEC task offloading in SCN. In [10], service delay is minimized by virtual machine (VM) mitigation in SCN. The impact of the mobility of UEs and the dense deployment of small cells have been discussed. The authors in [11] analyzed the coverage performance for computation tasks within the two-tier small-cell network. In [6], realizing the computation resource at one MEC-SBS (small cell base station) server is limited, a server collaboration scheme is designed where one UE could be served by multiple MEC-SBS servers simultaneously. Note that energy efficiency is not discussed in these works. Besides, the service delay requirement has been neglected so far.

Although there are some researches are dedicated to doing on MEC and SCN, it is just a start phase where they jointly consider these issues. Authors of [20] proposed a computational offloading with delay and capacity constraints problem in mobile edge computing. However, it only focused on an efficient design of a pre-fixed connected network with a set of small cells without considering fronthaul and backhaul links. In [21], we studied an energy cost minimization problem when offloading to multiple different small cells. Differently, in this paper, according to the characteristic of 5G SCN when offloading, we not only jointly consider the interference of the fronthaul and backhaul, but also consider the different computational capacity of each small cell. In addition, the delay constraint with considering a queue latency for subchannels allocated in practice is also considered. Finally, verify that our proposed AFSA-based scheme not only is applicable to large scale users, but also convergent to the global minimal.

III. System Model

In this paper, we consider a two-tier SCN integrated with MEC as shown in Fig. 1. The two-tier SCN is composed of a set of MBSs $\mathbb{N} = \{1, \ldots, N\}$ and FRSs $\mathbb{M} = \{1, \ldots, M\}$. For MBSs, they are connected to the core network via wired optical fibers and are installed with MEC servers to provide computation resources to UEs. We call MBSs with MEC servers as MEC-MBSs. For FRSs, they just serve as relays for UEs offloading their computation tasks to MEC-MBSs. For UEs $\mathbb{K} = \{1, \ldots, K\}$, in order to offload computation tasks to MEC-MBSs, they first send their tasks through the fronthaul link (wireless link from UEs to FRSs) to neighboring FRSs. Then these tasks are further relayed to MEC-MBSs [22]. Assume that each UE has a computation task $L_i = \{b_i, w_i, t_i^{\max}\}, i \in \mathbb{K}$ to accomplish. Here, b_i represents the size of the task, w_i is the total computation resource requirement (i.e., CPU cycles per unit time) to complete the task, and $t_i^{\max}$ is the maximal tolerable delay experienced by this task.

In this research, we adopt frequency band Ω_1 in the fronthaul link, frequency band Ω_2 in the backhaul link. The physical layer channel access scheme i.e., CDMA is adopted to allow multiple users of one base station to share the same spectrum resource simultaneously and efficiently. We aim to minimize the overall energy cost from all system entities, including UEs, FRSs and MEC-MBSs, during task offloading. Since both task transmission and computation consumes energy [23], we jointly consider these two aspects in our problem formulation. In the following, we first introduce the communication model and computation task processing model in details.

A. Communication Model

The wireless connections between UEs and MEC-MBSs consist of fronthaul links and backhaul links.

For the fronthaul link that connects UE $i \in \mathbb{K}$ to FRS $m \in \mathbb{M}$, its achievable data rate is calculated by

$$r_{i,m} = W \log_2 \left(1 + \frac{p_{i,m} g_{i,m}}{\sigma^2 + \sum_{l=1}^{M} \sum_{j=1, j \neq i}^{K} p_{j,l} g_{j,m}} \right) \quad (1)$$

where W is the channel bandwidth, $p_{i,m}$ is UE i's transmission power over its fronthaul link to FRS m, $g_{i,m}$ denotes the channel gain between UE i and FRS m, and σ^2 is the Gaussian noise. The interference to FRS m comes from concurrent transmissions of other fronthaul links occupying the same channel.

Similarly, the achievable data transmission rate over the backhaul link from FRS $m \in \mathbb{M}$ to MEC-MBS $n \in \mathbb{N}$ is formulated as

$$r_{m,n} = W \log_2 \left(1 + \frac{p_{m,n} g_{m,n}}{\sigma^2 + \sum_{q=1}^{N} \sum_{l=1, l \neq m}^{M} p_{l,q} g_{l,n}} \right) \quad (2)$$

where $p_{m,n}$ is the transmission power of FRS m over its backhaul link to MEC-MBS n and $g_{m,n}$ denotes the channel gain between these two. The interference suffered by MEC-MBS n comes from the neighboring MEC-MBSs.

B. Computation Task Processing Model

We then introduce the computation task processing model for task offloading. The task processing model includes task transmission and task computation.

$a_{i,m}$ denotes the access decision indicator, which is a binary variable. $a_{i,m} = 1$ represents UE i's task is connected to FRS m; Otherwise, $a_{i,m} = 0$. Denote by $c_{m,n}$ the computation task offloading decision indicator. $c_{m,n} = 1$ means that FRS m offloads the task to MEC-MBS n; otherwise, $c_{m,n} = 0$.

Let Δ_i be the total latency experienced by UE i's task. It is calculated by

$$\Delta_i = \Delta_{i,m}^t + Q^i + \Delta_{m,n}^t + \Delta_n^c. \quad (3)$$

where $\Delta_{i,m}^t$ means the transmission latency when UE i transmits the task to FRS m, and is calculated by $\Delta_{i,m}^t = b_i / r_{i,m}$. Q^i represents a waiting time when UE i's task in FRS m forms a queue and waits for subchannels allocated from MEC-MBS. This waiting time Q^i stands for the queuing delay experience in accessing the limited number of subchannels available at MEC-MBS. Consider that there are $\mathbb{B}$ subchannels available at MEC-MBS to the contention to transmit their computational offloading tasks to MEC-MBS. The total number of subchannels is $\mathbb{B}$ in SCN and we model the 5G system using M/M/$\mathbb{B}$ queue to analyze average subchannels allocation delay performance. It can be calculated by $Q^i = b_i / \lambda$, where λ denotes the average rate of subchannels allocation in the backhaul link [24], [25]. $\Delta_{m,n}^t$ represents the transmission latency when transmitting task from FRS m to MEC-MBS n, which is calculated by $\Delta_{m,n}^t = \sum_{i=1}^{K} a_{i,m} b_i / r_{m,n}$. Δ_n^c denotes the latency for computing the task on MEC-MBS n,

$$\Delta_n^c = \frac{\sum_{i=1}^{K} a_{i,m} w_i}{f_0} \quad (4)$$

where f_0 is the available computation resource, in terms of CPU cycles per unit time, provided by MEC servers at MEC-MBS n.

Denote by $e_{i,m}^t$ the energy cost when UE i transmits task to FRS m, we have $e_{i,m}^t = p_{i,m} \Delta_{i,m}^t$. The energy cost of FRS m offloading tasks to MEC-MBS n can be formulated as

$$e_{m,n} = p_{m,n} \Delta_{m,n}^t + \varepsilon_1 \sum_{i=1}^{K} a_{i,m} w_i \quad (5)$$

where ε_1 is the energy cost of the MEC servers for running one CPU cycle.

C. Problem Formulation

To minimize the total energy cost of the system, we formulate an energy optimization problem below

$$
\begin{aligned}
\min_{a_{i,m},c_{m,n}} \quad & \sum_{m=1}^{M}\left(\sum_{i=1}^{K} a_{i,m} e_{i,m}^{t} + \sum_{n=1}^{N} c_{m,n} e_{m,n}\right)\\
\text{s.t. } C_1 : & \sum_{m=1}^{M} c_{m,n} \sum_{i=1}^{K} a_{i,m} w_i \le w_n^{\text{total}}\\
C_2 : & \sum_{i=1}^{K} a_{i,m} r_{i,m} = c_{m,n} r_{m,n}\\
C_3 : & \sum_{m=1}^{M} a_{i,m} \le 1\\
C_4 : & \sum_{n=1}^{N} c_{m,n} = 1\\
C_5 : & \Delta_i \le t_i^{\max}\\
C_6 : & a_{i,m}, c_{m,n} \in \{0,1\}.
\end{aligned} \tag{6}
$$

Here, C_1 requires that the total CPU cycles provided by MEC-MBS n should not be larger than its maximum available computation resources w_n^{total}. C_2 means that the total incoming data rate at FRS m should be equal to its outgoing data. C_3 states that each UE i can connect to at the most one FRS. C_4 states that each FRS m is allowed to select only one MEC-MBS for task computation. C_5 ensures that the latency of UE i meets its maximum latency requirement.

IV. AFSA-Based Computation Offloading Algorithm

Since the above formulated optimization problem is a binary non-linear programming problem, it is NP-complete. Thus, it is desirable to develop an efficient algorithm to solve it. There have been quite a few existing meta-heuristic algorithms to solve NP-hard problems, such as Ant Colony Optimization (ACO), Tabu Search (TS), Genetic Algorithm (GA), Simulated Annealing (SA) and Artificial Fish Swarm Algorithm (AFSA). In particular, AFSA has a mechanism to enable the searching behavior to jump out of local extreme points and obtain the global optimization solution. Therefore, in this research we adopt AFSA to solve the energy optimization problem.

AFSA has been widely used in parameter estimation, robust PID parameter tuning, neural network training, and combinatorial optimization. It is an intelligent and stochastic swarm algorithm [26]. By following fish's behaviors, each AF intends to find the most optimal position by searching the solution space. Typically, AFSA has four key elements which are “Fish Behavior”, “Fitness function”, “Position” and “Solution Space”, respectively. In the process of searching, AF changes its position by moving to another direction in an evolutionary way, and the evolution continues until the value generated by fitness function reaches optimum. Searching efficiency highly depends on fish behaviors. Three fish behaviors, prey behavior, swarm behavior and following behavior, can make AFSA less sensitive to initial values to achieve the global optimization with stronger robustness [27]. The characteristics of fish behaviors are discussed as follows.

Prey behavior: This behavior enables the AF to choose the position at each evolution step randomly and freely, which also enriches the diversity of positions and fits for the evolution law of biological community. The mathematical model of prey behavior is as follows. Suppose that the current position and fitness value of the ith artificial fish are X_i and Y_i, respectively. If $Y_i < Y_{\text{new}}$, then AF evolves to a new position X_{new} following the rule below

$$X_{\text{new}} = X_i + Rand() \cdot \Theta \tag{7}$$

where $Rand()$ is a function of the random number and Θ is the solution space.

Swarm behavior: Swarm behavior urges AFs to group together spontaneously so as to avoid harm. As a result, the small number of AFs that are trapped into local optimal positions have a chance to jump out of local optimal positions. This behavior can be achieved by the following mathematical model. Assume the position of fish swarm center as X_center and the number of AFs in one swarm as n_f. Then we have

$$X\_center = \frac{\sum_{i=1}^{n_f} X_i}{n_f}. \tag{8}$$

If there are many “nutrient food” in the fish swarm center and they are not too crowded, i.e., function $Y\_center > Y_i$ and $Y\_center/(n_f \cdot Y_i) > \delta$ $(0 < \delta < 1)$, where δ is a congestion factor [28], the AF would move one further step to the central position. Then the new position is updated by

$$X_{\text{new}} = X_i + Rand() \cdot step() \cdot \frac{X\_center - X_i}{\|X\_center - X_i\|} \tag{9}$$

where $step()$ denotes the adjustment step function.

Following behavior: When one AF finds an optimal region with enough food and is not too crowded, the nearby AFs will follow it and reach its nearby position fast. Following behavior accelerates the move speed of AF toward a better position. In other words, it helps to speed up the convergence of AFSA by directing AFs following others' behaviors. The mathematical model can be described as

$$X_{\text{new}} = X_i + Rand() \cdot step() \cdot \frac{X\_\text{MAX} - X_i}{\|X\_\text{MAX} - X_i\|} \tag{10}$$

where X_MAX denotes the position of AF with maximum food density.

We then develop our scheme based on AFSA to solve our energy optimization problem. Still, two main issues should be stressed, i.e.,

1) How to describe the position and solution space of the AF?
2) How to define the fitness function to evaluate the quality of the specified position that one AF chooses?

Since the behaviors of all individuals are independent, real-time and parallel, we assume that potential computation offloading decisions of all the UEs construct one position for the AF. Let the computation offloading decision for one UE be $x_i = (a_{i,m}, c_{m,n})$. One position of the AF can be denoted as $X = \{x_1 \ldots, x_K\}$. Assume the solution space as Θ, $X^z \in \Theta, z = \{1, \ldots, Z\}$, especially, $Z = ((M+1) \times N)^K$. The size of the solution space is depended on the visual scope $Visual$, i.e., we use norm to measure the distance of one AF and other AFs. When the distance between two AFs is in the visual scope, we can get one solution space Θ. The solution space can be regarded as the environment where AF lives in. A swarm of AFs move around in the solution space to look for the global optimum position (or solution) that provides better quality.

Whether the position is better can be evaluated by the fitness function, which is denoted as

$$Fitness = \sum_{m=1}^{M} \left(\sum_{i=1}^{K} a_{i,m} e_{i,m}^{t} + \sum_{n=1}^{N} c_{m,n} e_{m,n} \right) + Penalty(a, c) \tag{11}$$

where $Penalty(a, c)$ shown below denotes the penalty function. It provides an efficient way to guide the AF to escape from the non-feasible region quickly.

$$\begin{aligned} Penalty(a,c) &= h(a,c) + g(a) \\ h(a,c) &= \sum_{m=1}^{M} \sum_{n=1}^{N} h_1 \times \left(c_{m,n} r_{m,n} - \sum_{i=1}^{K} a_{i,m} r_{i,m} \right)^2 \\ &\quad + \sum_{m=1}^{M} h_2 \times \left(\sum_{n=1}^{N} c_{m,n} - 1 \right)^2 \\ g(a) &= \sum_{n=1}^{N} \left(g_1 \times \left(\frac{1}{\sum_{m=1}^{M} c_{m,n} \sum_{i=1}^{K} a_{i,m} w_i - w_n^{\text{total}}} \right)^2 \right) \\ &\quad + \sum_{i=1}^{K} \left(g_2 \times \left(\frac{1}{\sum_{m=1}^{M} a_{i,m} - 1} \right)^2 \right) \\ &\quad + \sum_{i=1}^{K} \left(g_3 \times \left(\frac{1}{\Delta_i - t_i^{\max}} \right)^2 \right) \end{aligned} \tag{12}$$

where h_1, h_2, g_1, g_2 and g_3 are the penalty factors. The penalty factors are the weights of inequality constraint functions in the original constrained problem (6). Specially, h_1 and h_2 are the weights of constraints C_2 and C_4. g_1 and g_2 and g_3 are the weights of constraints C_1 and C_3 and C_5. They are to ensure that the AFSA is executed within the feasible region. Then the inequality constraint functions with the corresponding weights are combined with objective function in the constrained problem (6) to get the new objective function (11). So the constrained problem (6) is transformed to an unconstrained one. The optimal solution of the unconstrained problem (11) is consistent with that of the constrained problem (6).

Algorithm 1: AFSA-based computation offloading algorithm.

```
Initialization:
UE set: $\mathbb{K} = \{1, \ldots, K\}$ ;
One position for the AF: $X = \{x_1 \ldots, x_K\}$;
The number of AFs: $fish\_num$;
$j$th position for AF: $X^j = \{x_1^j \ldots, x_K^j\}$;
Fitness function value: $Fitness(X^j)$;
The maximum of searching times: $try\_number$;
The visual scope of AF: $Visual$;
The solution space of AF : $\Theta$;
Length in one move: $step$;
Congestion factor: $\delta$;
Maximum iterations: MAX_GEN;
Iteration times: $gen\_times = 1$;
while $gen\_times <$ MAX_GEN do
    for $j = 1 : fish\_num$ do
        $Fitness(X^j\_prey) = Algorithm2$
        $(X^j, step, \delta, Visual)$
        $Fitness(X^j\_swarm) = Algorithm3$
        $(X^j, step, \delta,$
        $Visual)$
        $Fitness(X^j\_follow) = Algorithm4$
        $(X^j, step, \delta,$
        $Visual)$
    end for
    $Fitness(X^{best}) = min\{Fitness(X^j\_prey),$
    $Fitness(X^j\_swarm), Fitness(X^j\_follow)\}$
    $gen\_times + +$;
end while
Output: $Fitness(X^{best})$
```

We assume there are $fish\_num$ AFs in the solution space Θ. In Algorithm 1, after initializing the parameters, each AF performs prey behavior, swarm behavior, and following behavior iteratively until the optimal position X^{best} is identified with the optimal fitness value. The optimal fitness value can be determined by the following function

$$Fitness(X^{best}) = min\{Fitness(X^j\_prey), Fitness(X^j\_swarm), Fitness(X^j\_follow)\}. \tag{13}$$

The previous algorithms are that the AFs need to explore the current environment to decide one behavior. However, in order to reduce the error of environmental judgment in this paper, we design an AFSA-based computation offloading algorithm (ACOA) to implement the three behaviors parallel to achieve the optimal fitness value. So the optimal fitness value is generated by comparing the fitness value calculated by prey behavior, swarm behavior and following behavior. Next, we will introduce prey behavior algorithm, swarm behavior algorithm and following behavior algorithm respectively. The implementation of ACOA is shown as Fig. 2.

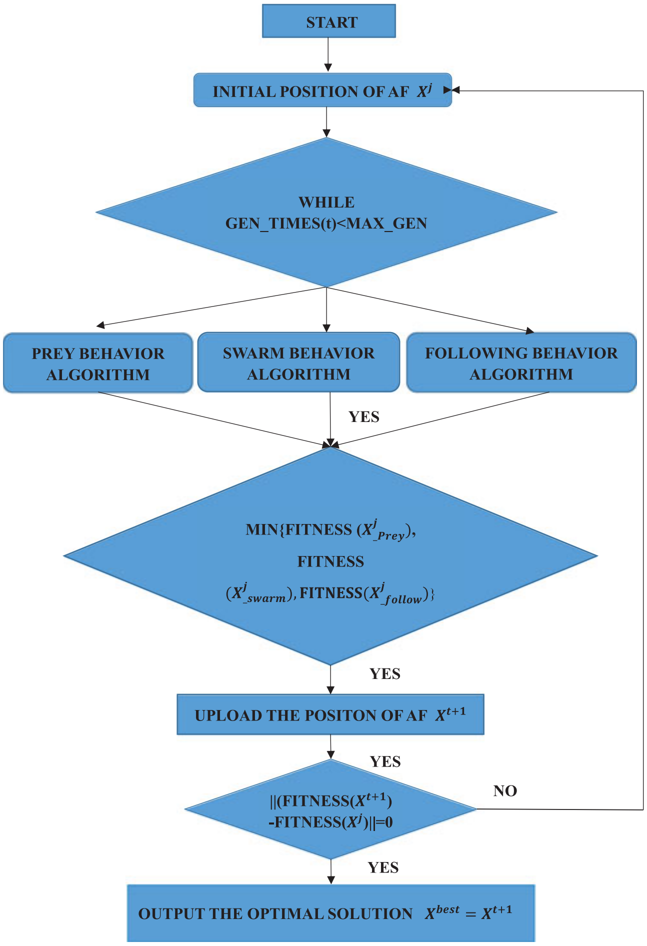
START
INITIAL POSITION OF AF X^j
WHILE
GEN_TIMES(t)<MAX_GEN
PREY BEHAVIOR ALGORITHM
SWARM BEHAVIOR ALGORITHM
FOLLOWING BEHAVIOR ALGORITHM
YES
MIN{FITNESS ($X^j_{\_Prey}$),
FITNESS
($X^j_{\_swarm}$), FITNESS($X^j_{\_follow}$)}
YES
UPLOAD THE POSITON OF AF X^{t+1}
YES
NO
‖(FITNESS(X^{t+1})
-FITNESS(X^j)‖=0
YES
OUTPUT THE OPTIMAL SOLUTION $X^{best} = X^{t+1}$

Fig. 2. The flow chart of AFSA-based computation offloading algorithm.

1) Prey Behavior Algorithm: Prey behavior is an individual behavior that each AF performs without considering other swarm members. When performing this behavior, each AF attempts try_number times around itself to achieve a better position [28], Which is often a local optimal position.

With UEs' current position X^j, solution space Θ and maximum searching number try_number, prey behavior algorithm is shown in Algorithm 2. If $Fitness(X^{t+1}) < Fitness(X^j)$, the AF will execute $X^{t+1} = X^j + floor(Rand(0,1) \times step \times (X^{t+1} - X^j)/\| X^{t+1} - X^j \|)$ to choose X^{t+1} as the new position, otherwise, it searches for a new position in the solution space randomly following step 7. This algorithm continues until the searching times achieve the maximum value or the fitness value of AF in position X^{t+1} is the same with that in X^j.

2) Swarm Behavior: When we use the prey behavior algorithm to solve the energy minimization problem, UEs may be caught in a local range. Under this context, swarm behavior should be considered. According to congestion factor δ, we take swarm behavior algorithm to let AF jump out of local best optimal value and speed up the convergence speed.

With AF's current position X^j, solution space Θ, maximum searching times try_number, the neighboring AF swarm, swarm behavior algorithm are shown in Algorithm 3.

Algorithm 2: Prey behavior algorithm.

```
    Input: $X^j, Visual, step, \delta$
2:  Initialization:
    Searching times: t = 0;
    Current searching position $X^t = X^j$:
    Current fitness value: $Fitness(X^t) = Fitness(X^j)$;
    for $t = 1 : try\_number$ do
4:      $X^{t+1} = X^t + floor(Rand(0,1) \times Visual)$
        if $Fitness(X^{t+1}) < Fitness(X^t)$ then
6:          $X^{t+1} = X^t + floor(Rand(0,1) \times step \times \frac{X^{t+1}-X^t}{\|X^{t+1}-X^t\|})$,
        else if $Fitness(X^{t+1}) > Fitness(X^t)$ then
8:          $X^{t+1} = X^t + floor(Rand(0,1) \times step)$
        end if
10:     $X^t = X^{t+1}$, $Fitness(X^t) = Fitness(X^{t+1})$
    end for
12: $\|Fitness(X^{t+1}) - Fitness(X^t)\| = 0$
    Output: $X^t$ and $Fitness(X^t)$.
```

Algorithm 3: Swarm behavior algorithm.

```
    Input: $X^j, Visual, step, \delta$
    Initialization:
    Searching time: t = 0;
    Current searching position $X^t = X^j$:
    Current fitness value: $Fitness(X^t) = Fitness(X^j)$;
    The number of neighborhoods in the AF swarm: $n_f$;
    The center position of the swarm center: $X\_center$;
3:  while $t = 1 : try\_number$ do
        $n_f = |Z(X^t, Visual)|$
        $X\_center = Center(Z(X^t, \Theta))$
6:      if $(\frac{Fitness(X\_center)}{n_f Fitness(X^t)} < \delta)$ then
            $X^{t+1} = X^t + floor(Rand(0,1) \times step \times \frac{X\_center-X^t}{\|X\_center-X^t\|})$
        else if $\frac{Fitness(X\_center)}{n_f Fitness(X^t)} > \delta$ then
9:          $X^{t+1} = X^t + floor(Rand(0,1) \times step)$
        end if
        $X^t = X^{t+1}$, $Fitness(X^t) = Fitness(X^{t+1})$;
12: end while
    $\|Fitness(X^{t+1}) - Fitness(X^t)\| = 0$
    Output: $X^t$ and $Fitness(X^t)$.
```

Specifically, n_f denotes the number of AFs in the neighboring AF swarm, X_center represents the center position of the neighboring swarm, $X\_center = Center(Z(X^j, Visual))$, $Center(\ Z(\ X^j, Visual)) = (X^1 + X^2 + \cdots + X^{fish\_num})/n_f$, and δ denotes the congestion factor. If $Fitness(X\_center)/n_f < \delta Fitness(X^j)$, which means the position of center can produce higher fitness value and this place is not too crowded, the AF can evolve closer to the center and the next position of AF can be calculated by function $X^{t+1} = X^j + floor(Rand(0,1) \times step \times (X\_center -$

Algorithm 4: Following behavior algorithm.

```
Input: $X^j, Visual, step, \delta$;
Initialization:
Searching time: t = 0;
Current searching position $X^t = X^j$:
Current fitness value: $Fitness(X^t) = Fitness(X^j)$;
 The number of neighborhoods in the AF swarm: $n_f$;
The position of the $k$th neighborhood for the AF:
 $X^k, k = \{1, ..., Z\}$;
for $t = 1 : try\_number$ do
    $n_f = |Z(X\_min, Visual)|, X\_min \in Z(X^t, Visual)$;
    $Finess(X\_min) = MIN(Fitness(X^k))$;
    if $(Fitness(X\_min)/n_f < \delta Fitness(X^j))$ then
        $X^{t+1} = X^t + floor(Rand(0,1) \times step \times \frac{X\_min - X^t}{\|X\_min - X^t\|})$;
    else if $(Fitness(X\_min)/n_f > \delta Fitness(X^t))$ then
        $X^{t+1} = X^t + floor(Rand(0,1) \times step)$;
    end if
    $X^t = X^{t+1}, Fitness(X^t) = Fitness(X^{t+1})$;
end for
$\|Fitness(X^{t+1}) - Fitness(X^t)\| = 0$
Output: $X^t$ and $Fitness(X^t)$.
```

$X^j)/\| X\_center - X^j \|)$. Otherwise, the algorithm will jump to move one step randomly. Since our problem is a binary non-linear programming problem, we also introduce a rounding function $floor(x)$, which trunk x to its closest integer no larger than it.

3) Following Behavior: With following behavior, one AF has the capacity of observing and learning others. It speeds up the convergence of our algorithm. We describe its details in Algorithm 4.

AF with current position X^j follows the behavior of neighboring AF with position X_min, which can be figured out by $X\_min = arg\, MIN(Fitness(X^k))$. If $Fitness (X\_min)/n_f < \delta Fitness(X^j), (0 < \delta < 1)$ is satisfied, then AF will move towards to X_min and the next position can be determined by $X^{t+1} = X^j + floor(Rand(0,1) \times step \times (X\_min - X^j)/\| X\_min - X^j \|)$. In addition, if we find that the fitness value satisfies $Fitness(X\_min)/n_f > \delta Fitness(X^j), (0 < \delta < 1)$, the AFs will move one step randomly. The Algorithm 4 will terminate until the condition 13 is met.

V. Global Convergence Analysis

In this section, we use finite Markov chain to analyze the convergence properties of the AFSA based scheme. Firstly, since the solution space of the scheme is discrete, we can model the transition process for AFs from one position to another as a finite Markov chain. Secondly, if the transition matrix of Markov chain is reducible, stochastic and stable, the global convergence of the proposed AFSA based scheme can be proven.

We first introduce transition process model based on the finite Markov chain. A finite Markov chain describes a probabilistic trajectory over a finite sate space S. We define the finite and discrete total state space F of cardinality $|F| = |Z|$. According to the execution of AFSA based scheme, $S = \{fitness(x)|x \in F\}$ is the actual search state space, where $|S| = fish\_num \times (3 * try\_number) \times \mathrm{MAX\_GEN}$, obviously, $|S| < |F|$. Then we divide the state space F into $F = \{F^1, \ldots, F^{|S|}\}$. The transition probability $p_{i,j}(t)$ shows that the AF transfers from state $i \in |S|$ to state $j \in |S|$ at step t. The transition probabilities of a finite Markov chain can be gathered in a transition matrix $P = (p_{i,j})$, where $p_{i,j} \in [0,1]$ and $\sum_{j=1}^{|S|} p_{i,j} = 1$ for all $i \in |S|$ (or $\sum_{i=1}^{|S|} p_{i,j} = 1$ for all $j \in |S|$). Specially, X_k^j $(k = 1, 2, \ldots, |S|, j = 1, 2, \ldots, |F^k|)$ denotes jth position state of the AF k. If an AF k moves from state X_k^j to state X_l^m, we denote the transition probability by $p_{kj,lm}$, where $kj \in S$ and $lm \in S$. According to the characteristic of Markov chain, we can get

$$p_{kj,l} = \sum_{m=1}^{|F^k|} p_{kj,lm} \tag{14}$$

$$\sum_{l=1}^{|S|} p_{kj,l} = 1 \tag{15}$$

$$p_{k,l} \geq p_{kj,l} \tag{16}$$

Then we introduce several different properties of Markov chain as follows:

- Property 1: The Markov chain is positive. If $a_{i,j}$ of a square matrix $A_{|S| \times |S|}$ satisfies $a_{i,j} > 0$ for all $i, j \in \{1, ..., |S|\}$, the transition matrix of Markov chain is positive.
- Property 2: The Markov chain is primitive. If a square matrix $A_{|S| \times |S|}$ is nonnegative (i.e., $A_{|S| \times |S|} \geq 0$), the transition matrix of Markov chain is primitive.
- Property 3: The Markov chain is reducible. If there exists a $k \in N$ such that A^k is positive, and A can be brought into the form (specially, C and T are also square matrices)

$$\begin{pmatrix} C \ldots 0 \\ R \ldots T \end{pmatrix} \tag{17}$$

  the transition matrix of Markov chain is reducible.
- Property 4: The Markov chain is stochastic. If $a_{i,j}$ of a square matrix $A_{|S| \times |S|}$ satisfies $\sum_{j=1}^{|S|} a_{i,j} = 1$ for all $i \in \{1, \ldots, |S|\}$, the transition matrix of Markov chain is stochastic.
- Property 5: The Markov chain is stable. If a stochastic matrix $A_{|S| \times |S|}$ has identical rows, the transition matrix of Markov chain is stable.

Based on the properties of Markov chain, we introduce Theorem 1. The two properties of Theorem 1 are also the theoretical basis of the convergence of AFSA based scheme.

Theorem 1: If the transition matrix P of a Markov chain is a reducible stochastic matrix, and it has the following two properties: (1) P^k converges as $k \to \infty$ to a positive stable stochastic matrix P^∞, where $\lim_{k\to\infty} P^k = P^0 \cdot \lim_{k\to\infty} P^k = P^0 \cdot P^\infty$, and we can get that P^∞ is a stable stochastic matrix, unique regardless of the initial distribution, and has nonzero entries. (2) The reducible stochastic matrix P will exist a $m \times m$ primitive stochastic matrix C, and $R, T \neq 0.$, we can obtain:

$$P^\infty = \lim_{k\to\infty} P^k = \lim_{k\to\infty} \begin{pmatrix} C^k \ldots 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} \ldots T^k \end{pmatrix} \tag{18}$$

after simplifying the matrix:

$$P^\infty = \begin{pmatrix} C^\infty \ldots 0 \\ R^\infty \ldots T \end{pmatrix} \tag{19}$$

where P^∞ satisfy the following conditions:

$$p^\infty = (p_{i,j})_{|S|\times|S|} = \begin{cases} p_{i,j} > 0, 1 \leq i \leq |S|, 1 \leq j \leq m. \\ p_{i,j} = 0, 1 \leq i \leq |S|, m \leq j \leq |S|. \end{cases} \tag{20}$$

According to properties of Markov chain and Theorem 1, we will prove the convergence of AFSA based scheme.

Theorem 2: AFSA based scheme has the property of the global convergence.

Proof: In order to prove the global convergence property of AFSA based scheme, we have the following three steps. We first demonstrate the transition matrix P of the AFSA based scheme is positive. Then, there exist $C, R, T, 0$ matrices, and the transition matrix can be decomposed into $C, R, T, 0$ matrices. Finally, according to the two properties of Theorem 1, the global convergence of AFSA based scheme is proven.

Firstly, it is obvious that the transition matrix P of the AFSA based scheme with the transition probability $p_{i,j} \in [0, 1]$ is positive.

Secondly, we decompose the transition matrix P into $C, R, T, 0$ matrices according to Lemma 1. Since Lemma 1 has been proven successfully in [29], we will not give detailed proof in this paper.

Lemma 1: In AFSA, $\forall X_k^j \in X^j, k = 1, 2, \ldots, |S|, j = 1, 2, \ldots, |F^k|$, satisfy:

$$\forall i > k, p_{k,i} = 0 \tag{21}$$

$$\exists i < k, p_{k,i} > 0 \tag{22}$$

According to the Lemma 1, we can get the transition matrix P as follows:

$$P = \begin{pmatrix} p_{1,1} & 0\ldots & 0 \\ p_{2,1} & p_{2,2}\cdots & 0 \\ \vdots & \vdots & \vdots \\ p_{|S|,1} & p_{|S|,2}\cdots & p_{|S|,|S|} \end{pmatrix} = \begin{pmatrix} C \ldots 0 \\ R \ldots T \end{pmatrix}. \tag{23}$$

Specially, we set $C = (p_{1,1}) = 1 \neq 0$, and $R = (p_{2,1}, p_{3,1}, \ldots, p_{|S|,1})^T$, matrix T is as follows:

$$T = \begin{pmatrix} p_{2,2}\cdots & 0 \\ \vdots & \vdots \\ p_{|S|,2}\cdots & p_{|S|,|S|} \end{pmatrix} \tag{24}$$

Thirdly, according to the definition of reducible stochastic matrix, we know that transition matrix P is a reducible stochastic matrix. Then, based on the stochastic property of Markov chain and Theorem 1, we know that C^∞ is equal to 1, i.e., $C^\infty = 1$, and P^∞ is a stable stochastic matrix, unique regardless of the initial distribution, and has nonzero entries, so $R^\infty = (1, 1, \ldots, 1)^T$. Thus

$$P^\infty = \begin{pmatrix} 1 & 0\ldots & 0 \\ 1 & 0\ldots & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0\ldots & 0 \end{pmatrix} \tag{25}$$

We can get $\lim_{t\to\infty} P\{Fitness(X^t) = Fitness(X^{best})\} = 1$, where $Fitness(X^{best})$ represents the best fitness within the solution space by the state X^{best} at step $best$, and $Fitness(X^{best})$ is the global optimum of energy optimization problem. Based on the analysis above, the convergence of AFSA based scheme can be proven.

VI. Numerical Results

In this section, simulation results of the proposed ACOA are presented in comparison with other algorithms. The simulation is conducted on a Matlab-based simulator 2016. The simulation parameters are described as follows. We consider the system consisting of 25 UEs, 5 FRSs and 3 MEC-MBSs [30], [31]. The distance between FRSs and UEs are all $l = 50$ m. The distance between FRSs and MEC-MBS are all $l = 1$ km. The wireless total bandwidth of each base station (MBS/FRS) is $W = 5$ MHz. The transmission power is ranged from $p = 50$ mW to 100 mW randomly. The background noise is $\sigma^2 = -100$ dbm. The energy cost of the MEC servers on the MBSs is $\varepsilon_1 = 4$ J/GHz. According to the wireless channel model for cellular radio environment, we set channel gain as $g = l^{-\varsigma}$, where ς is the path loss factor and $\varsigma = 4$. The Data size of computation tasks b_i is randomly distributed between 5 Mbit and 20 Mbit [32]. We set the number of CPU cycles required by UEs randomly distributed between 0.1 GHz and 2 GHz. Without loss of generality, we assume that each MEC has the same computing capability as $f_0 \in (1, 10)$.

To evaluate the impact of different parameters in ACOA, we next implement the simulations with three parameters (e.g., $Visual, \delta$ and $step$). We first present the convergence of the proposed ACOA with different values of parameter $Visual$. As shown in Fig. 3, when the $Visual$ is in the visual scope between 1 and 3, with an increasing the number of the iteration, the energy cost declines dramatically first and then enters a

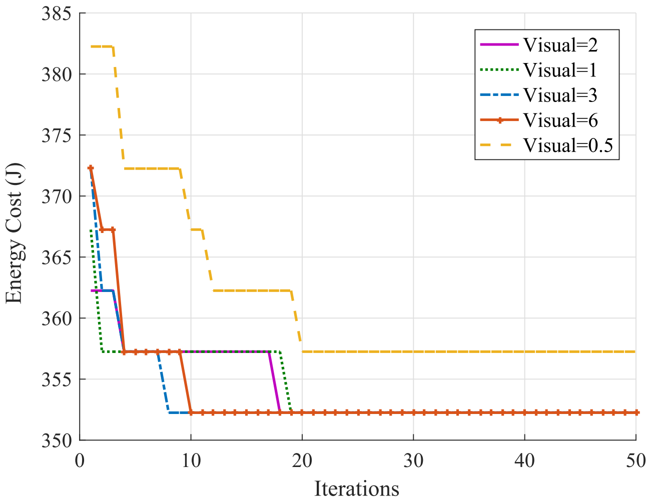

Fig. 3. Energy cost of the system versus different $Visual$.

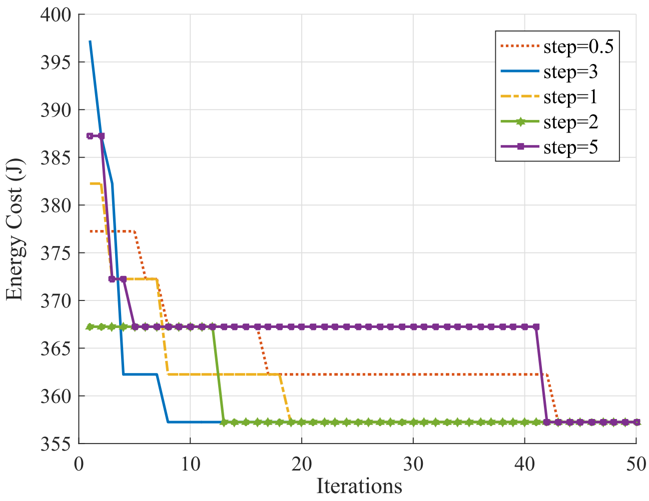

Fig. 5. Energy cost of the system with different searching $step$.

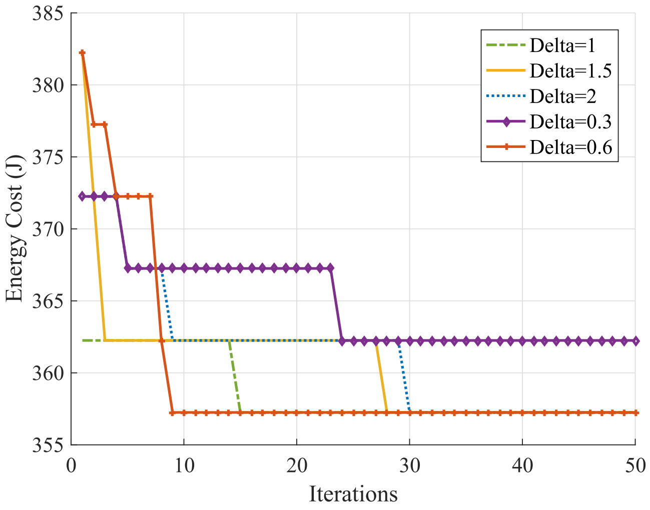

Fig. 4. Energy cost of the system with different δ.

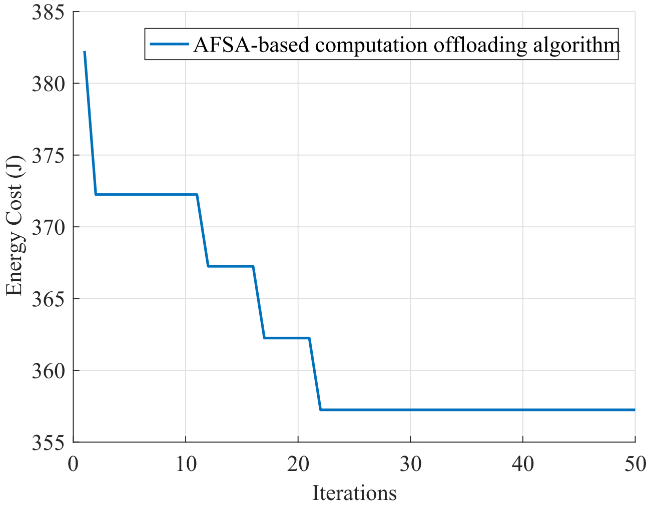

Fig. 6. The convergence of the AFSA-based computation offloading algorithm.

stable status within the first 20 iterations. This is due to the fact that the parameter of $Visual$ is optimized, the advantage over optimal solution is significant. When the $Visual = 0.5$, the yellow line shows the searching process may be into the local optimum, which is because visual scope of AF is small, the feasible solutions in the solution space become few as well. However, when $Visual = 6$, the number of the iteration is lager than $Visual = 3$, it is because the lager visual scope is, the more AF is, the more feasible solutions are.

We plot the energy cost with respect to the increasing number of iterations about parameter δ in Fig. 4. The simulation is conducted under different δ's. With all these values, our algorithm converges to the same energy cost except $\delta = 0.3$. When $\delta = 0.3$, the number of the iteration is the largest and it can not converge to the minimum. It is because that the smaller the congestion parameter δ is, the smaller AFs are in the visual scope. During the range of δ from 0.6 to 2, we can see that the energy cost rapidly decreases as the number of the iteration increases and then it enters a stable status. Because the value of parameter of δ is adaptive to search the minimum. Thus we choose $\delta = 0.6$ as the parameter of ACOA. The lines within this figure also demonstrate that the smaller value of parameter δ leads to a better network performance.

In Fig. 5, we show the energy cost of the offloading system with different searching length $step$. As the number of iterations increase, the energy cost of the system reaches a stationary state with different $step$. And when $step$ in the scope between 0.5 and 3, we can see that as the value of parameter $step$ increases, the iteration number declines. This is because when the searching $\{$step$\}$ becomes larger, the optimal solution can be identified faster.However, when the $step = 5$, the number of the iteration becomes large but ACOA can converge to the minimum finally. It demonstrates that the parameter of $step$ is not the bigger the better. According to the result, we choose $step = 3$ as the value of parameter $step$ for our proposed ACOA.

In Fig. 6, we discuss the convergence property of ACOA over three parameters $Visual$, δ and $step$. We set $Visual = 3, \delta =$

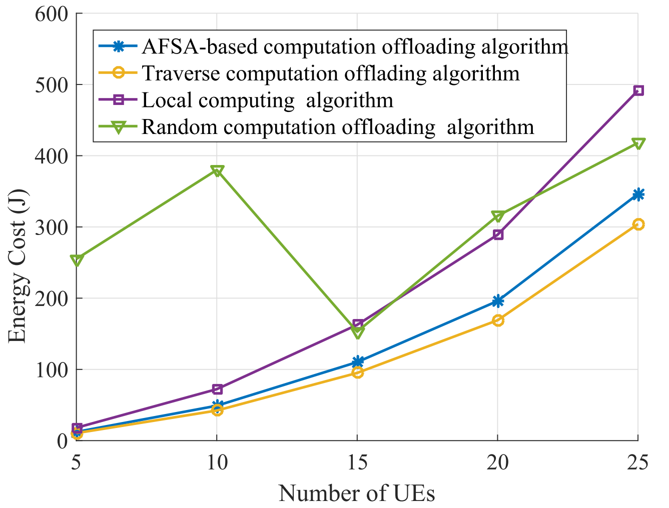

Fig. 7. Energy cost versus the different number of UEs.

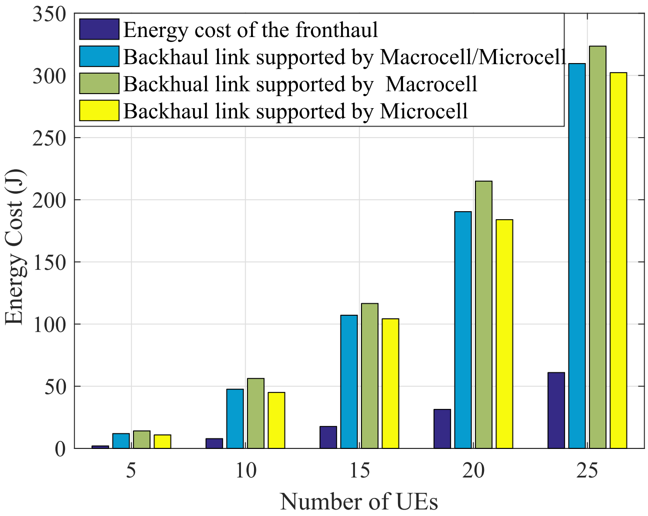

Fig. 8. Energy cost of the fronthaul link and the backhaul link.

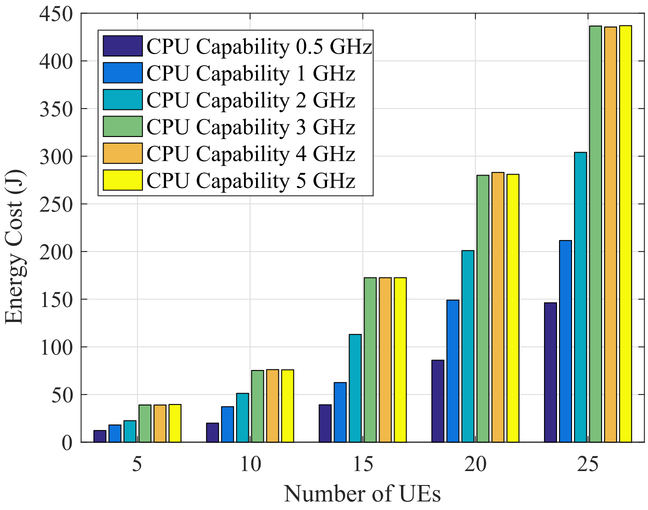

Fig. 9. Energy cost of the system with the different MEC server CPU capabilities.

0.6, $step = 3$. We can see that the proposed algorithm can keep mobile devices' energy cost decreasing and then converge to a stable solution. Besides, the iteration number to achieve the convergence is less than 25.

In Fig. 7, we show the energy cost generated by different computation offloading schemes. They are the proposed AFSA-based computation offloading algorithm, the traverse computation offloading algorithm, the random computation offloading algorithm and local computing algorithm [14]. The traverse computation offloading algorithm goes through every kind of computation offloading solution and selects the one that minimizes the system energy cost. The random computation offloading algorithm assigns one FRS and one MEC-MBS to one UE. It can be seen that as the number of UEs increases, the energy cost also increases. We also notice that the energy cost of the ACOA is a little higher than that of the traverse offloading algorithm. This is because ACOA searches part of the solution space. For the traverse computation offloading algorithm, the whole solution space is checked. However, the complexity of the traverse computation offloading algorithm is much higher than that of the ACOA. We also observe that the energy cost of the random computation offloading algorithm and that of the local computing algorithm vary randomly. Because they do not have mechanisms to ensure the system performance, their performances are worse than ACOA.

Fig. 8 shows the energy cost of the fronthaul link and the backhaul link under different number of UEs in the two-tier SCN integrated with MEC. The figure demonstrates that the energy cost of the fronthaul link is far less than that of the backhaul link. This is because the distance between UEs and FRSs is longer than that between FRSs and MEC-MBSs. The figure also shows that the reasonable deployment of the backhaul link is of great importance for energy saving. Thus, in the condition that we ensure the topology structure of the fronthaul link is unchanged, we only change the topology structure of the backhaul link. We thus run the simulations with three scenarios in the backhaul link, which are the backhaul link supported by Macrocell/Microcell, Macrocell and Microcell, respectively. In the first scenario, we deploy one Macrocell BS and two Microcell BSs in the backhaul link and each BS is deployed one MEC server. In the second scenario, we only deploy three Macrocell BSs deployed three MEC servers to support the backhaul link. In the last scenario, we deploy three Microcell BSs deployed three MEC servers for the backhaul link. Fig. 8 shows that the energy cost increases as an increasing the number of UEs. It is also observed that the energy cost of the second scenario is higher than that of the third scenario. However, the gap between the first scenario and the third scenario is narrow. The third scenario can save 10% energy cost compared with the second one. Since the operation and maintenance cost of Macrocell BSs is very high, we know that it is more wise to deploy Microcell BSs for the backhaul link.

Fig. 9 shows that the energy cost versus the different MEC server CPU capacities. Apparently, under the same computation

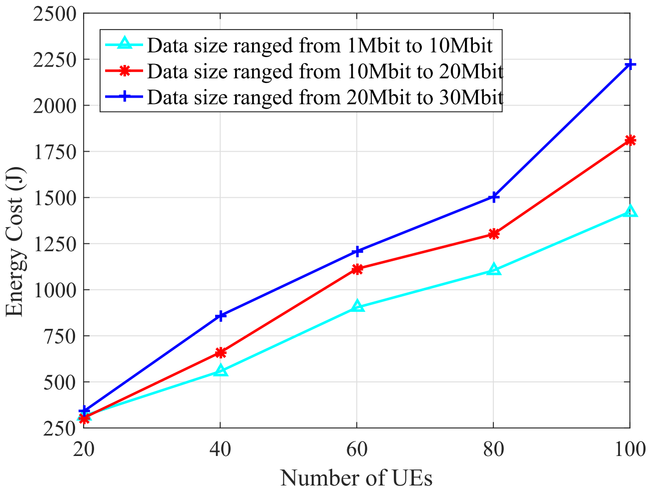

Fig. 10. Sum energy cost versus the size of the offloading data.

capacity, when the the number of UEs becomes larger, the energy cost increases dramatically. This is because when the number of UEs who choose computation offloading increases, the computation tasks to be executed also grows, so the energy cost of the system is increasing. Besides, under the same number of UEs, when the computation capacity of the MEC-MBS improves, the energy cost of the system increases. It is because the higher computation capacity is, the more computation tasks are executed. As a result, the network energy cost generated by task computation grow.

Fig. 10 shows that the sum energy cost of the proposed scheme versus the size of the offloading data. In Fig. 10, we can see that with the size of offloading data increasing, the energy cost grows as well. The main reason is that the size of the offloading data increases, so the transmission delay is augmented, and the increased energy cost is caused by the process of computation tasks transmission. In addition, it is worth noting that the increasing trend of the sum energy cost of different size of offloading data is rapid as the number of UEs increases. One increment of energy cost is caused by the process of computation tasks transmission. The other is mainly because the interference among UEs become severe when the number of UEs grows, and this leads to a relatively high sum energy cost.

VII. Conclusion

In this paper, we investigate energy efficient task offloading in 5G MEC. We focus on a two-tier small-cell network scenario. An energy minimization problem is formulated by jointly considering the energy cost at both the fronthaul link and the backhaul link. Energy consumption at both task transmissions and computations are modeled. To efficiently solve the energy cost minimization problem, we develop an algorithm that can achieve global convergence. Simulation results demonstrate the efficiency and effectiveness of the proposed algorithm.

References

[1] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, “Enabling personalized search over encrypted outsourced data with efficiency improvement,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[2] S. Wang, Y. Zhang, H. Wang, Z. Wang, X. Wang, and T. Jiang, “Large scale measurement and analytics on social groups of device-to-device sharing in mobile social networks,” *Springer Mobile Netw. Appl.*, vol. 23, no. 4, pp. 1–13, 2017.

[3] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, “D2d big data: Content deliveries over wireless device-to-device sharing in realistic large scale mobile networks,” *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 1–10, Feb. 2018.

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

[5] J. Zhang, W. Xie, F. Yang, and Q. Bi, “Mobile edge computing and field trial results for 5g low latency scenario,” *Wireless Commun. Over Zigbee Autom. Inclination Meas. China Commun.*, vol. 13, no. Suppl. 2, pp. 174–182, 2017.

[6] I. Ketyk, L. Kecsks, C. Nemes, and L. Farkas, “Multi-user computation offloading as multiple knapsack problem for 5g mobile edge computing,” in *Proc. 2016 Eur. Conf. Netw.Commun.*, Jun. 2016, pp. 225–229.

[7] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges,” *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[8] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: The role of MEC in the internet of things,” *IEEE Consumer Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Sep. 2016.

[9] R. Al-Zaidi, J. Woods, M. Al-Khalidi, K. M. Alheeti, and K. McDonald-Maier, “Next generation marine data networks in an IoT environment,” in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput.*, Jun. 2017, pp. 8–11.

[10] Z. Yan, W. Zhou, S. Chen, and H. Liu, “Modeling and analysis of two-tier hetnets with cognitive small cells,” *IEEE Access*, vol. 5, pp. 2904–2912, 2017.

[11] M. Deng, H. Tian, and X. Lyu, “Adaptive sequential offloading game for multi-cell mobile edge computing,” in *Proc. 2016 23rd Int. Conf. Telecommun.*, May 2016, pp. 1–5.

[12] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, “Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control,” *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[13] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge-clouds,” in *Proc. 2015 IFIP Netw. Conf.*, May 2015, pp. 1–9.

[14] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE ACM. Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[15] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Trans. Signal Inform. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[16] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, “Energy aware offloading for competing users on a shared communication channel,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 87–96, Jan. 2017.

[17] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, “Joint allocation of computation and communication resources in multiuser mobile cloud computing,” in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun.*, Sep. 2013.

[18] M.-H. Chen, B. Liang, and M. Dong, “Joint offloading decision and resource allocation for multi-user multi-task mobile cloud,” in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 22–27.

[19] Y. Cao, C. Long, and T. Jiang, “Share communication and computation resources on mobile devices: A social awareness perspective,” *IEEE Wireless Commun.*, vol. 23, no. 4, Aug. 2016.

[20] W. Wang and W. Zhou, “Computational offloading with delay and capacity constraints in mobile edge,” in *Proc. 2017 IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.

[21] H. Zhang, J. Guo, L. Yang, X. Li, and H. Ji, “Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC,” in *Proc. 2017 IEEE Conf. Comput. Commun. Workshops*, May 2017, pp. 115–120.

[22] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5g: challenges, methodologies, and directions," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 78–85, May 2016.

[23] M. Li and P. Li, "Crowdsourcing in cyber-physical systems: Stochastic optimization with strong stability," *IEEE Trans. Emerging Topics Comput.*, vol. 1, no. 2, pp. 218–231, Dec. 2013.

[24] M. Patra, R. Thakur, and C. S. R. Murthy, "Improving delay and energy efficiency of vehicular networks using mobile femto access points," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1496–1505, Feb. 2017.

[25] T. Aktas, G. Quer, T. Javidi, and R. R. Rao, "From connected vehicles to mobile relays: Enhanced wireless infrastructure for smarter cities," in *Proc. 2016 IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–6.

[26] Y. Luo, J. Zhang, and X. Li, "The optimization of pid controller parameters based on artificial fish swarm algorithm," in *Proc. IEEE Int. Conf. Autom. Logist.*, Oct. 2007, pp. 1058–1062.

[27] X. Liu, P. Liu, and H. Zheng, "Distribution network structure planning based on QAFSA," in *Proc. 2012 China Int. Conf. Elect. Distrib.*, Apr. 2012, pp. 1–4.

[28] X. Li, Y. Xue, F. Lu, and G. Tian, "Parameter estimation method based-on artificial fish school algorithm," *J. Shandong Univ. (Eng. Sci.)*, vol. 34, no. 3, pp. 1672–3961, Jun. 2014.

[29] G. Huang, J. Liu, and Y. Yao, "Artificial fish algorithm global convergence is proved," *Comput. Eng.*, vol. 38, no. 2, pp. 1000–3428, Jan. 2012.

[30] D. Wu, Q. Wu, Y. Xu, and Y. C. Liang, "Qoe and energy aware resource allocation in small cell networks with power selection, load management, and channel allocation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7461–7473, Aug. 2017.

[31] Y. Guo, Q. Yang, J. Liu, and K. S. Kwak, "Cross-layer rate control and resource allocation in spectrum-sharing OFDMA small-cell networks with delay constraints," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4133–4147, May 2017.

[32] M. M. Mowla, I. Ahmad, D. Habibi, and Q. V. Phung, "An energy efficient resource management and planning system for 5g networks," in *Proc. 2017 14th IEEE Annu. Consumer Commun. Netw. Conf.*, Jan. 2017, pp. 216–224.

Lichao Yang received the B.S. degree in mathematics and statistics from Zhengzhou University, Henan, China, in 2015. She is currently working toward the Ph.D. degree in the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include 5G small cell networks, fog and edge computing, and online multiple-dimension resource management.

Heli Zhang (M'18) received the B.S. degree in communication engineering from Central South University, Changsha, China, in 2009, and the Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. Since 2014, she has been a Lecturer with the School of Information and Communication Engineering, BUPT. She has been the reviewer for various journals such as IEEE Wireless Communications, IEEE Communication Magazine, IEEE Transactions on Vehicular Technology, IEEE Communication Letters, and IEEE Transactions on Networking. She participated in many National projects funded by National Science and Technology Major Project, National 863 High-tech, and National Natural Science Foundation of China, and cooperated with many Corporations in research. Her research interests include heterogeneous networks, long-term evolution/fifth generation, and Internet of Things.

Ming Li (M'10) received the B.E. degree in electrical engineering from Sun Yat-sen University, Guangzhou, China, in 2007, the M.E. degree in electrical engineering from Beijing University of Posts and Communications, Beijing, China, in 2010, and the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2014. She is currently an Assistant Professor with the Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA. Her research interests include wireless security, privacy-preserving data analysis, resource management, and network optimization in cyber-physical systems and wireless networks.

Jun Guo received the B.S. degree from North China University of Technology, Beijing, China, in 2015. He is currently working toward the M.S. degree at Beijing University of Posts and Telecommunications, Beijing, China. His research interests include mobile edge computing and ultradense networks. During his two years' study at the university as a graduate student, his main focus was on the research of wireless communication of 5G. He has authored two papers on mobile edge computing in an international conference in 2017.

Hong Ji (SM'09) received the B.S. degree in communications engineering and the M.S. and Ph.D. degrees in information and communications engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1989, 1992, and 2002, respectively. In 2006, she was a visiting scholar with the University of British Columbia, Vancouver, BC, Canada. She is currently a Professor with BUPT. She has authored more than 200 journal/conference papers. Several of her papers had been selected for best paper. Her research interests include wireless networks and mobile systems, including ICT applications, system architectures, cloud computing, green communications, software defined networks, radio access, management algorithms, and performance evaluations. She is serving on the editorial boards of the IEEE Transaction on Green Communications and Networking and Wiley *International Journal of Communication Systems*. She has guest-edited Wiley *International Journal of Communication Systems*, special issue on "Mobile Internet: Content, Security and Terminal." She was the Co-Chair of Chinacom'11, and a member of the Technical Program Committee (TPC) of ISCIT'17, GC'17 Workshops, Globecom'16/15/14/13/12/11, ICC'13/12/11, IEEE VTC'12S, WCNC'15/12.