Mobility-aware Task Offloading and Migration Schemes in SCNs with Mobile Edge Computing

Zhaolin Liu, Xiaoxiang Wang, Dongyu Wang, Yanwen Lan, Junxu Hou
Key Lab. of Universal Wireless Comm., Ministry of Education, Beijing Univ. of Posts and Telecom., P.R. China
Email:zhaolin_l@163.com, dy_wang@bupt.edu.cn

***Abstract*—Mobile Edge Computing (MEC) is regarded as an enabling technology to provide computing capacity closer to the mobile equipments (MEs). Due to the MEs' mobility, the execution results of offloaded tasks can't be transmitted back to MEs if they are out of the coverage of the small cell network (SCN). Therefore, the migration is needed, which causes the extra cost. In this paper, a novel mobility-aware offloading and migration scheme with MEC is proposed to maximize the total revenue of MEs. Firstly, with the analysis of MEs' mobility, the problem of maximizing the total revenue of MEs is formulated, which integrates the offloading decision and resource allocation to reduce the probability of migration. Secondly, the problem is formulated as a mixed integer non-linear programming (MINLP) problem, so a sub-optimal algorithm named genetic algorithm based allocation algorithm (GAAA) is proposed to solve this NP-hard problem. Finally, extensive simulations demonstrate that GAAA can effectively improve the total revenue of MEs in contrast with other baseline algorithms.**

***Index Terms*—MEC, offloading, migration, mobility, genetic algorithm based allocation algorithm.**

I. Introduction

The rapid development of mobile communication has promoted a variety of new services emerging. With the evolution of the fifth-generation (5G) network technology, many innovative applications, such as augmented reality (AR), virtual reality (VR), Internet of Things (IoT) and Internet of Vehicles have been developed, most of which are closely related to the mobile equipments (MEs). However, due to the limited hardware condition, such as the limited computation capacity, storage size and power, the demands of these services for the network bandwidth, time-delay and so on cannot be satisfied. In recent years, mobile edge computing (MEC) technology, deployed at the edge of the mobile network, was proposed to solve the problem of energy consumption and time-delay.

One of the most important advantages of MEC is the ability to provide computation resource to MEs so as to reduce the local load. Therefore, resource management has become a significant research point. In the multiple small cell networks (SCNs), MEs can be served by many small base stations (SBSs). How to choose the appropriate one has been studied in [1] and [2]. Paper [1] propose an adaptive sequential offloading game approach to solve the offloading problems and adjusted the number of offloading MEs in order to reduce the queuing delay. In paper [2], the proposed framework jointly considers the content caching and full duplex communication to provide high-data-rate service. An alternating direction method of multipliers algorithm is adopted to optimize the user association, power control and resource allocation. Dividing the services into content caching and computing is also considered in [3]. It introduces a new concept of task caching which caches the completed task application and their related data in MEC server. Then the alternating iterative algorithm is proposed to jointly optimize the task caching and offloading in order to minimize the energy consumption. The different conditions of sub-channels is also an important influence factor and paper [4] jointly optimizes the offloading decision, resource allocation and the sub-channel allocation with a distributed joint computation offloading and resource allocation optimization scheme. The authors in [5] and [6] assume that the tasks can be divided into several pieces and study the computation offloading and content caching in wireless block chain networks with MEC. The tasks can be executed either on the MEC server or in a group of device-to-device (D2D) users.

The papers mentioned above rarely consider the mobility of MEs. Due to the mobility, MEs will eventually leave the coverage of a SCN. In this case, the results of offloaded tasks will be migrated to a new SBS which is able to serve the MEs [7]. In [8], a shared MEC is proposed in order to reduce the handover and migration times. Dividing a metropolitan area into disjoint MEC regions according to the movements of users is studied in [9] to minimize the number of possible migrations. And paper [10] considers an ultra-dense network with many SBSs. The tasks are portioned and offloaded to many SBSs at different locations due to the mobility. A novel energy-aware user-centric mobility management scheme is developed to choose the SBSs.

However, this paper aims at reducing the probability of migration to maximize the total revenue of MEs by optimizing the offloading decision and computation resource allocation. The distinct contributions of this paper are as follows:

- A novel mobility-aware offloading and migration scheme is proposed in SCNs with MEC. Since leaving

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

the cell without the offloaded tasks finished will cause the migration of the execution results. The offloading and computation resource allocation are jointly optimized based on the MEs' mobility to reduce the probability of migration, which maximizes the total revenue of MEs.

- The mobility in this paper refers to the sojourn time which is formulated with an exponential distribution. The factor of the distribution represents the average sojourn time varying from different MEs.
- A utility function considering the price and revenue coefficients is derived to model the system. So the problem is formulated as a mixed integer non-linear programming (MINLP) problem, which is also an NP-hard problem. The genetic algorithm based allocation algorithm (GAAA) is adopted to obtain the suboptimal solution.
- Extensive simulations show the performance of GAAA compared with the other baseline algorithms.

The rest of this paper is organized as follows. In Section II, we propose the framework of the small cell network with MEC considering the MEs' mobility and formulate the problem. In Section III, we present the suboptimal algorithm GAAA to solve the problem. Section IV analyzes the simulation and discusses the results. Finally, Section V draws the conclusion of the work.

II. System Framework

As shown in Fig. 1, the MEC server is deployed at SBS, which is located in the center of the cell, and MEs are randomly distributed in the cell. The radius of a SCN is assumed to be relatively small so that MEs won't stay in the cell for a long time because of the mobility. In Fig. 1, $ME1$ represents the MEs which are still in the coverage of SCN when the offloaded tasks are finished on the MEC server. $ME2$ refers to the MEs which have left the cell before the tasks are finished. In this case, the execution results of the tasks will be migrated to another SBS, the process of which causes the extra cost. Let $N_u = \{1, 2, ..., i, ...N\}$ represent the set of MEs in SCN. The task of ME i can be described in three terms as $W_i = \{M_i, f_i, T^{\max}\}, \forall i \in N_u$, where M_i is the size of the computational task, f_i is the required computation resource, and $T^{\max}$ is the maximum latency. The number of CPU cycles is used to measure the computation resource required for the tasks [11]. This paper assumes that the size of the computation task is large and the task is computation-intensive. Therefore, offloading tasks can save amount of time and energy consumption. However, due to the limited computation resource on MEC server, which is denoted as C_{MEC}, not all the tasks can be offloaded when the number of MEs is large. Denote $A = \{a_i, i \in N_u\}$ as the set of the offloading decision. Let $a_i = 1$ when the task is offloaded, and $a_i = 0$ means that the task will be operated locally.

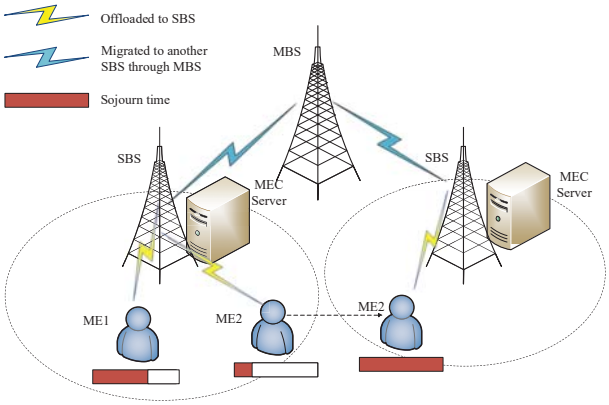


Fig. 1. An illustration of the SCNs Architecture with MEC considering the mobility

A. Network Model

Assume that the transmission power of ME i is a constant for simplicity, denoted by p_i. And θ is used for the standard path loss propagation exponent. With the distance between ME i and SBS, represented by d_i, the signal-to-noise ratio (SNR) of ME i can be expressed as

$$SNR_i = \frac{p_i h_i d_i^{-\theta}}{\sigma^2}, \forall i \in N_u \tag{1}$$

where σ^2 is the power of additive white Gaussian noise and h_i represents the channel gain. Therefore, the uploading transmission rate of ME i can be expressed as $R_i = W_0 \log_2(1 + SNR_i), \forall i \in N_u$.

B. Computation Model

In this part, the computation model is proposed to show the time and energy consumption of the tasks executed locally or on the MEC server.

1) Local Computation: Set C_i^{local} as the local computational capability (CPU cycles) of ME i. So the local execution time can be given as

$$t_i^{local} = \frac{f_i}{C_i^{local}}, \forall i \in N_u. \tag{2}$$

According to [12], the local energy consumption is expressed as

$$E_i^{local} = \kappa f_i (C_i^{local})^2, \forall i \in N_u \tag{3}$$

where κ is the effective switched capacitance relying on the chip architecture [13].

2) Edge Computation: When the tasks are offloaded to the MEC server, it includes two parts, transmission and execution. According to the parameters above, the transmission time can be given as $t_i^{ul} = M_i / R_i$. And the energy consumption is $E_i^{ul} = p_i t_i^{ul}$. Let $C = \{C_i, i \in N_u\}$ be

the MEC computation capability allocation profile, where C_i means the computation resource allocated to ME i. So the execution time can be expressed as $t_i^{exe} = f_i/C_i, \forall i \in N_u$. It should be noted that because of the limited computation resource, the resource assigned to the MEs must satisfy the following constraint $\sum_{i=1}^{N} C_i \leq C_{MEC}$. Compared with the input data size, the output data is so small that can be ignored [14]. Therefore, the total edge computation time is derived as

$$t_i^{MEC} = t_i^{ul} + t_i^{exe}, \forall i \in N_u \tag{4}$$

C. Mobility Model

In this paper, let the sojourn time represent the mobility of MEs and model it by an exponential function [8], [15]. Therefore, the probability density function (PDF) of sojourn time, denoted by $f_{\tau_i}(t)$, is given as

$$f_{\tau_i}(t) = \frac{1}{\tau_i} e^{-\frac{t}{\tau_i}}, t \geq 0 \tag{5}$$

where τ_i represents the average sojourn time of ME i. Due to the MEs' different characteristics, τ_i varies among them. For simplicity, assume that τ_i follows the gaussian distribution here. But in reality, the reliable τ_i can be obtained by collecting the information of MEs through machine learning tools.

D. Problem Formulation

In this section, a utility function is derived to describe the cost and revenue of MEs when the tasks are offloaded to the MEC server. Adopt the time saving and local energy saving as the revenue. The occupied resource and migration are considered as the cost. Time saving refers to the subtraction between the local execution time and edge computation time, which is expressed as

$$T_i^{gain} = \gamma(t_i^{local} - t_i^{MEC}), \forall i \in N_u \tag{6}$$

where γ is the coefficient of time saving revenue. The resource cost include the energy consumption of transmission and MEC computation resource. So we can get the total resource cost expressed as

$$Z_i^{MEC} = \alpha E_i^{ul} + \beta C_i, \forall i \in N_u \tag{7}$$

where α is the unit price of energy consumption and β is the unit price of computation resource.

When the tasks are offloaded, according to the relationship of size between the edge computation time and the predicted sojourn time of MEs, the utility function is discussed in two different cases.

Case 1: The tasks can be finished on the MEC server before MEs leaving the cell, which can be expressed as $t_i^{MEC} < t$, where t is the sojourn time. In this case, migration won't happen. Since the sojourn time t follows the exponential distribution, according to (5), the probability of case 1 can be denoted by $P_{\tau_i}(t > t_i^{MEC})$. And the utility function in this case is derived as

$$Q_i^1 = T_i^{gain} + Z_i^{\text{local}} - Z_i^{MEC}, \forall i \in N_u \tag{8}$$

where $Z_i^{\text{local}} = \alpha E_i^{local}$ is the local energy saving revenue.

Case 2: When the tasks are finished on the MEC server, MEs have already left the cell, the probability of which is $P_{\tau_i}(t \leq t_i^{MEC})$. In this case, the results will firstly be migrated to the SBS in another SCN where the MEs are going and then transmitted back to the MEs, the process of which will cause the extra cost, denoted as Z_i^{mig}. To make it easy to deal with, the migration cost is assumed to be only related to the size of the task, given as $Z_i^{mig} = \delta M_i$ [8]. In reality, besides the size, the cost is also related to the type of the task and so on [7]. So the utility function of this case can be expressed as

$$Q_i^2 = T_i^{gain} + Z_i^{local} - Z_i^{MEC} - Z_i^{mig}, \forall i \in N_u \tag{9}$$

Combine (8) and (9), we can get the expression as

$$Q_i = \begin{cases} Q_i^1 & , P_{\tau_i}(t > t_i^{MEC}) \\ Q_i^2 & , P_{\tau_i}(t \leq t_i^{MEC}) \end{cases}, \forall i \in N_u \tag{10}$$

Therefore, the expectation of the revenue of ME i is used to measure the performance, which is derived as

$$\begin{aligned} \overline{Q}_i = & P_{\tau_i}(t > t_i^{MEC}) Q_i^1 \\ & + P_{\tau_i}(t \leq t_i^{MEC}) Q_i^2, \forall i \in N_u \end{aligned} \tag{11}$$

$$P_{\tau_i}(t \leq t_i^{MEC}) = \int_0^{t_i^{MEC}} \frac{1}{\tau_i} e^{-\frac{t}{\tau_i}} dt = -e^{-\frac{t_i^{MEC}}{\tau_i}} + 1 \tag{12}$$

$$P_{\tau_i}(t > t_i^{MEC}) = 1 - P_{\tau_i}(t \leq t_i^{MEC}) \tag{13}$$

Specially, when $a_i = 0$, the tasks will be executed locally and MEs cannot get any revenue from the MEC server. So let $\overline{Q}_i = 0$ in this case.

As a result, the optimization problem can be formulated with offloading decision and computation resource allocation as follows

$$\begin{aligned} \max_{A,C} \ & \sum_{i \in N_u} a_i \overline{Q}_i(C_i) \\ s.t. \ & C1: \ t_i^{ul} < t_i^{MEC} < T_i^{\max}, \ \forall i \in N_u \\ & C2: \ \sum_{i \in N_u} C_i \leq C_{MEC} \\ & C3: \ a_i \in \{0, 1\}, \ \forall i \in N_u \\ & C4: \ a_i = I(C_i), \ \forall i \in N_u \end{aligned} \tag{14}$$

In (14), constraint C1 ensures that the edge computation time won't exceed the maximum latency and the computation resource won't be negative. C2 constrains that the quantity of computation resource distributed to the MEs won't exceed the total computation resource of MEC. While C3 limits the range of the variable a_i. In C4, $I(\cdot)$ is the indicator function. If $\cdot > 0$, let $I(\cdot) = 1$; otherwise, set $I(\cdot) = 0$. C4 ensures that the computation resource won't be allocated to the ME which decides to execute locally.

III. Proposed Algorithm

As aforementioned, the problem of (14) is a MINLP problem, which is also an NP-hard problem. Therefore, a suboptimal algorithm called genetic algorithm based allocation algorithm (GAAA) is implemented to solve this optimization problem because of its better global search property. This section mainly discusses how to use GAAA to obtain the suboptimal solution of (14), as shown in Algorithm 1.

Firstly, the utility function is regarded as the fitness function to evaluate the goodness of the individuals. The constraints of the problem will be ensured within the initialization and selection.

Secondly, since the optimization problem requires high precision, the real coded strings are chosen as the chromosomes. Each one of the chromosome is a solution of problem (14), which is represented by

$$L_i^{MEC} = [L_1, ...L_i..., L_N] \tag{15}$$

where $L_i = [a_i, C_i]^{\mathrm{T}}$ is the set of the variables of ME i. According to the constraint C4 of (14), if $a_i = 0$, let $L_i = [0, 0]^{\mathrm{T}}$.

Algorithm 1 The Genetic Algorithm based Allocation Algorithm

Input: N_u, K, P_c, P_m, T.
Output: L_{best}, Q_{best}.

1: **Initialize:** Set K individuals for the population in a random way under the constraints of (14). Calculate the fitness value of each individual and sort out the biggest one as Q_{best}. Set the best individual as L_{best}.
2: **for** $t = 1$ to T **do**
3: Randomly choose two individuals and do the crossover operation with the probability P_c. The crossover operation applies the uniform crossover operation for the set A and the recombination method for the set C;
4: Select the individuals from the parents and offsprings for mutation operation with the probability P_m;
5: Calculate the fitness value of each new individual and divide them into the feasible ones and infeasible ones;
6: Do the random tournament selection operation and make sure the best individual is sorted out. Compare the best individual of iteration t, denoted by L_{best}^t, with the historical best individual L_{best}. If L_{best}^t is better than L_{best}, then let $L_{best} = L_{best}^t$ and renew Q_{best};
7: **end for**

Then as for the selection operator, the random tournament selection is applied because of its lower computation complexity and better selectivity of good individuals [16]. Each time two individuals are randomly selected and the better one is saved. It won't stop until the number of individuals comes to the initial population size. In case of the best individual being ignored during the selection operation, the worst one of the next generation is sorted out and replaced with the best individual.

Next, considering the crossover operator, the set A and C are operated independently with the probability P_c because of their different encoding methods. For the set A, we use uniform crossover operation. It can accelerate the convergence and prevent from falling into the local extremum. For C, we use the recombination method, which can be expressed as

$$\begin{aligned} C_i\,(child1) &= (1-b)\,C_i\,(parent1) + b\,C_i\,(parent2) \\ C_i\,(child2) &= b\,C_i\,(parent1) + (1-b)\,C_i\,(parent2) \end{aligned} \tag{16}$$

where b is a random variable between $(0, 1)$.

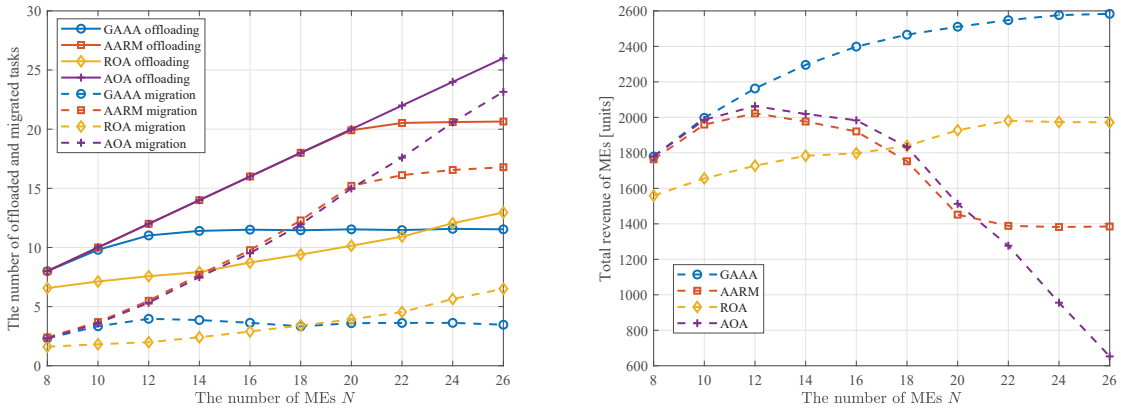
At last, the mutation operation is also different from the set A to C. Exchange 0 and 1 for a_i and add or subtract a random variable to C_i with the probability P_m within the constraints of (14). Select the feasible ones first and add with the infeasible ones which are near the boundary. Because after the next iteration the infeasible ones are more likely to become feasible.

IV. Simulation Results And Discussions

In this section, according to [2] and [4], the parameters are set as follows. The coverage radius of SCN is $80\,m$ and MEs are randomly distributed. The transmission power of MEs follows the Gaussian distribution $CN(\mu_1, \sigma_1^2)$, where $\mu_1 = 20\,dBm$, and $\sigma_1 = 2$. For channel parameters, the bandwidth is $10\,MHz$ and $\theta = 2$ is the path loss exponent. For the MEs, the local computation capacity follows the uniform distribution from 0.5 to $0.8\,GHz$. The size of the task is randomly assigned at the range of $[100, 150]\,MB$ and $f_i = \varepsilon M_i$ is the required computation resource, where ε is randomly set from $[0.4, 0.5]\,Gcycles/MB$. Due to the large size of the task, the maximum latency is $T_i^{\max} = 80\,s$. The price of the different types are α= $8\,units/Joule$, β=$2\,units/GHz$ and $\gamma = 4\,units/second$. According to [13], we set $\kappa = 10^{-11}$. As for the mobility, the average sojourn time of MEs follows the Gaussian distribution $CN(\mu_2, \sigma_2^2)$, where $\mu_2 = 40\,seconds$ and $\sigma_2 = 20$. As for the parameters of GA, $K = 32$, $P_c = 0.6$ and $P_m = 0.1$.

The performance of the proposed Algorithm GAAA is compared with the other three algorithms. Allocation algorithm regardless of the mobility (AARM) is similar with GAAA, except that it doesn't consider the mobility. Randomly offloading algorithm (ROA) and all offloading algorithm (AOA) both consider the mobility when allocating the resources. But ROA randomly offloads the tasks with the probability 0.5 and AOA offloads all the tasks.

Fig. 2(a) shows the number of offloaded tasks and migrated tasks with different algorithms versus the number of MEs. Fig. 2(b) shows the total revenue of MEs with different number of MEs. As the number of MEs increasing, the performance among them are quite different. Combine (a) and (b) in Fig. 2, we can conclude the reasons as follows. As for GAAA, the computation resource is adequate at first,



(a) (b)

Fig. 2. The impact of number of MEs, where total computation resource of MEC $C_{MEC} = 26\ GHz$ and migration cost $\delta = 0.5\ units/MB$ (a): The impact to the number of offloaded tasks and migrated tasks. (b): The impact to the total revenue of MEs.

so with the number of MEs increasing, more tasks can be offloaded and most of them can avoid migration. But when the number comes up to 12, both the number of offloaded tasks and migrated tasks begin to change slowly but the revenue continues to increase. Because the limited resource can only serve a certain number of MEs, but the larger number of MEs improves the diversity and selectivity. So the total revenue still increase but the rate is getting slower. The total revenue of AARM and AOA increase first because of the adequate resource. But with the number of migration becoming larger, the increment of migration cost will be bigger than the increment of gains. Then the total revenue begin to decrease until the number of MEs comes to 20. Both AOA and AARM offload all the tasks before that, but since AOA considers the mobility in resource allocation, its revenue is larger than AARM. Finally, AARM begins to keep stable because both the number of offloaded and migrated tasks are essentially unchanged. But in AOA, migration cost still increase and the time revenue begin to fall, so the total revenue still decrease and the rate becomes faster. The reason of ROA is similar with AOA, but due to the less number of offloaded tasks, it changes slowly.

The total revenue with different MEC computation resource is simulated in Fig. 3. It shows that the algorithms all get higher revenue with the growth of the total computation resource. But the rate of growth are different among them. The reasons are also related to the number of offloaded and migrated tasks. GAAA selectively offloads the tasks to reduce the probability of migration, so with the growth of the resource, it will guarantee that the offloaded tasks can be finished before leaving with high probability first, which causes the slowly increasing of the number of offloaded tasks. So the total revenue grows slowly. It is the same reason for ROA. The number of offloaded tasks is essentially

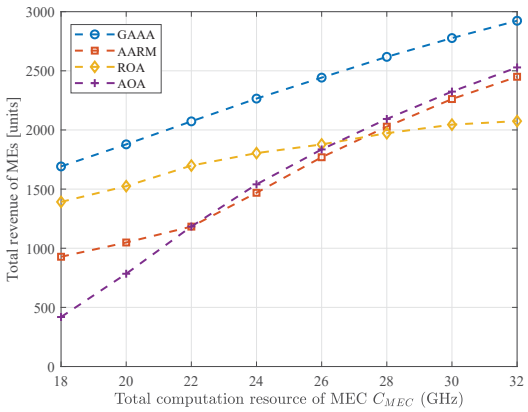


Fig. 3. The impact of the total computation resource of MEC C_{MEC} from 18 GHz to 32 GHz, where $N = 18$ and $\delta = 0.5\ units/MB$

unchanged and when the probability of migration is much lower, the increment of revenue is mainly related to the time gain, which is rather small. However, in AOA, although the offloaded tasks won't change, the migrated tasks becomes less and less, which saves amount of migration cost, so the total revenue increase fast. The reason for the revenue of AARM increasing so slowly at the beginning is that it doesn't consider the mobility. With the total resource increasing, it offloads more tasks instead of reducing the probability of migration. So the new offloaded tasks will be migrated with high probability and the increment of the revenue is little. It will last until all the tasks are offloaded. Then the rate will be the same as AOA. Therefore, Fig. 3 indirectly illustrates that the scheme which ignores the mobility will offload too many tasks and increase the number of migration.

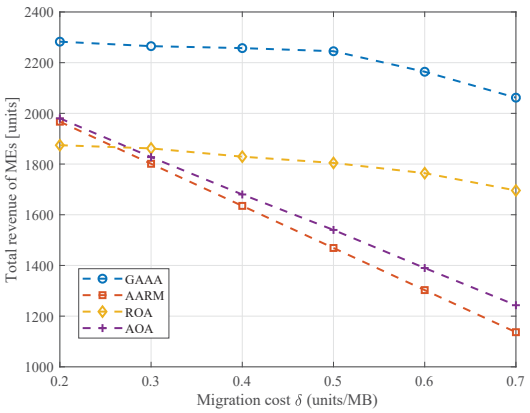


Fig. 4. The impact of the migration cost δ from 0.2 to 0.7 $units/MB$, where $N = 18$ and $C_{MEC} = 24\,GHz$

In Fig. 4, the total revenue are all decreasing with the increasing of migration cost. The slow speed of decline in GAAA illustrates that the proposed scheme in this paper can reduce the probability of migration significantly, so the increasing of migration cost has little impact on it. Due to the less offloaded tasks in ROA, the computation resource is relatively adequate, so the probability of migration is also small. In AARM and AOA, tasks are mostly offloaded, so the shortage of resource increases the probability of migration. Many tasks will be migrated and the total revenue is affected by the increasing of migration cost strongly. Specially, it can be observed that the revenue of AOA decreases slower than AARM. Because AOA considers the different mobility of MEs in resource allocation, which ensures that the probability of migration for some tasks can be lower.

V. Conclusion

In this paper, with the analysis of MEs' mobility, which refers to the sojourn time, the probability of migration is reduced by optimizing the offloading and computation allocation with MEC, which maximizes the total revenue of MEs. The problem is formulated as a MINLP problem. And a suboptimal algorithm named genetic algorithm based allocation algorithm (GAAA) is proposed to solve this NP-hard problem. The simulation results show that our proposed scheme can effectively save the migration cost and increase the total revenue of MEs. In the future work, the cooperative of multiple SBSs to reduce the migration will be considered.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61701038 and in part by the Fundamental Research Funds for the Central Universities.

References

[1] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *2016 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–5.

[2] Z. Tan, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Virtual resource allocation for heterogeneous services in full duplex-enabled scns with mobile edge computing and caching," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1794–1808, Feb 2018.

[3] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11 365–11 373, 2018.

[4] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19 324–19 337, 2018.

[5] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Joint computation offloading and content caching for wireless blockchain networks," in *IEEE Conference on Computer Communications Workshops*, April 2018, pp. 517–522.

[6] M. Liu, R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[7] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, February 2018.

[8] W. Nasrin and J. Xie, "Sharedmec: Sharing clouds to support user mobility in mobile edge computing," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[9] X. Guan, X. Wan, J. Wang, X. Ma, and G. Bai, "Mobility aware partition of mec regions in wireless metropolitan area networks," in *IEEE Conference on Computer Communications Workshops*, April 2018, pp. 1–2.

[10] Y. Sun, S. Zhou, and J. Xu, "Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, Nov 2017.

[11] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 398–401, June 2017.

[12] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[13] W. Hao and S. Yang, "Small cell cluster-based resource allocation for wireless backhaul in two-tier heterogeneous networks with massive mimo," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 509–523, Jan 2018.

[14] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

[15] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-aware coded probabilistic caching scheme for mec-enabled small cell networks," *IEEE Access*, vol. 5, pp. 17 824–17 833, 2017.

[16] M. V. O. D. Assis, A. H. Hamamoto, T. Abrao, and M. L. Proenca, "A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for dos/ddos mitigation on sdn networks," *IEEE Access*, vol. 5, pp. 9485–9496, 2017.