A Reinforcement learning-based for Multi-user Task Offloading and Resource Allocation in MEC

Tiange Xiang, Inwhee Joe*
Hanyang Univ, *Hanyang Univ.

singskyx1@hanyang.ac.kr, iwjoe@ hanyang.ac.

Abstract

Mobile edge computing (MEC), which enables mobile terminals to offload computational tasks to a server located at the user's edge, is considered an effective way to reduce the heavy computational burden and achieve efficient computational offloading. In this paper, we study a multi-user MEC system in which multiple user devices (UEs) can offload computation to the MEC server via a wireless channel. To solve the resource allocation and task offloading problem, we take the total cost of latency and energy consumption of all UEs as our optimization objective. To minimize the total cost of the considered MEC system, we propose an DRL-based method to solve the resource allocation problem in wireless MEC. Specifically, we propose a Asynchronous Advantage Actor-Critic (A3C)-based scheme. Asynchronous Advantage Actor-Critic (A3C) is applied to this framework and compared with DQN, and Double Q-Learning simulation results show that this scheme significantly reduces the total cost compared to other resource allocation schemes

I. Introduction

So far time delay and energy consumption are usually used as the measurement indexes for computing the performance of unloading [1]. According to the two performance standards of time delay and energy consumption, there are three solutions: the scheme of minimizing delay, the scheme of minimizing energy consumption and the scheme of minimizing the total cost [2] Solutions to minimize latency: Design a reasonable offload strategy to minimize latency on mobile devices. In Ref. [3] a low-complexity online Lyapunov optimization-based dynamic computation offloading (LODCO) algorithm is proposed. The LODCO algorithm makes an offload decision in each slot and then allocates CPU cycles to the UE (executed locally) or transmit power (offloaded to the MEC), which results in a 64% reduction in runtime. for energy minimization scheme In Ref. [4] A novel software defined edge cloudlet (SDEC) based RL optimization framework is proposed in this paper to tackle the energy minimization problem in wireless MEC. Specifically, Q-learning and cooperative Q-learning based RL schemes are proposed for the intractable problem. Simulation results reveal that the proposed scheme achieves superior performance in saving the battery power of a user device compared to other benchmark methods such as Q-learning with a random algorithm and Q-learning with epsilon greedy. total cost minimization scheme In Ref. [5] Specifically, the Q-learning based, and Deep Reinforcement Learning (DRL) based schemes are proposed. A weighted sum designed to minimize latency and energy consumption.

Motivated by the analysis, in this paper, we propose a computing offloading policy based on the A3C algorithm to minimize the sum cost. where each UE is composed of two deep neural networks: one is used as the function approximator to estimate the value functions in the critic part, and the other is used as a parameterized stochastic policy in the actor part. The multiple UE are trained asynchronously using policy gradient algorithm. From the final result, the proposed A3C algorithm outperforms the traditional DRL (DQN) algorithm In section III we propose an A3C-based theme-based theme in detail. In Section IV, we show the simulation results. Finally, the conclusion is drawn in Section V .

II. System Model

A. Network Model

A mobile edge offloading system is shown in Fig. 1 premeditate a single-cell scenario with an station and K UEs are represented as K= {1,2………K} each UE will generate only one computation-intensive task, Each UE could offload the task to the MEC server through wireless or execute it locally

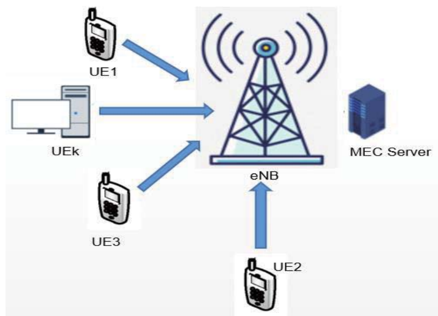

Fig1: Network Mode

Assume that each UE has an computing intensive task to be processed represented as $L_k(B_k, C_k, D_T)$. Here B_k is size of computation input data and C_k the total number of CPU cycles required to process the task L_k . D_T is the maximum latency tolerance. which means that the task execution time should not exceed D_T, whether the task is executed either locally or by computation offloading.

Moreover, we only concentrate on executing locally or offloading the task to the MEC server in this article, assume the task cannot be divided into partitions to be processed on different devices, which means that each UE should execute its task by local computing or offloading computing. We use

binary variables $a_k = \{0,1\}$to define offloading decisions where a_k=0 means the task is processed by UE 's CPU locally, and a_k=1 indicates the task is offloaded to the MEC server.

Define W as the bandwidth of wireless channel. There is only one eNB in one small cell, so the interval interference is neglected. It's assumed that if multiple UEs choose to offload the task simultaneously, the wireless bandwidth would be equally allocated to the offloading UEs for uploading data. the achievable upload data rate for UE k is

$$r_k = \frac{W}{K}\log\left(1+\frac{p_k * h_k}{\frac{W}{K}N0}\right) \quad (1)$$

where K is the number of offloading UEs, p_k is the transmission power of UE k for uploading data, h_k is the channel gain of UE k in the wireless channel, N0 is the variance of complex white Gaussian channel noise.

B. Task model and Calculation model:

If UE_k chooses to execute its task m locally, the local computing only includes the delay of offloading task to the CPU. here Tm the local computing delay Then f_k^l be the computation capability of the mobile device. The local execution delay of task m is T_k^l the computational energy *is* E_K^l Where Z_m represents the energy consumption per CPU cycle to complete task m We set $Z_k = 10^{-27}(f_k^l)^2$ according to the practical measurement proposed in [6]

$$T_k^l = \frac{C_k}{f_k^l} \quad (2)$$

$$E_k^l = Z_k * C_k \quad (3)$$

if UE chooses to offload the task to the MEC server for processing, then the total cost includes the delay and energy consumption of task upload and task processing. Set r_k as the data upload rate of the wireless channel when the subtask is uploaded, the calculation formula is as follows

$$T_k^{O\ Tran} = \frac{B_k}{r_k} \quad (4)$$

$$E_k^{O\ Tran} = p_k * T_k^{O\ Tran} = \frac{p_k * B_k}{r_k} \quad (5)$$

$$T_k^{O\ exe} = \frac{C_k}{f_k^o} \quad (6)$$

$$E_k^{O\ exe} = p_k^i * T_k^{O\ exe} = \frac{p_k^i * C_k}{f_k^o} \quad (7)$$

Where f_k^o is defined as the allocated computational resource When the task is executed on the MEC, UE k stays idle and define the power consumption of the idle state as p_k^i.

According to the (2–7), he total cost of all tasks in the system can be expressed as:

$$C_{all} = \sum_{k=1}^{K} a_k \left[I_k^t\left(T_k^{O\ Tran} + T_k^{O\ exe}\right) + I_k^e\left(E_k^{O\ Tran} + E_k^{O\ exe}\right)\right] + (1-a_k)\left(\left[I_k^t T_k^l + I_k^e E_k^l\right)\right. \quad (8)$$

Where I_k^t and I_k^e represent the weights of time and energy cost of task k. The weights satisfy $0 < I_k^t < 1$ $0 < I_k^e < 1$

C. Problem Formulation:

The objective of this paper is to minimize the sum cost of task offloading for all UEs, Under the constrain of maximum tolerable delay and computation capacity, the problem is formulated as follows:

$$\min \sum_{k=1}^{K} C_{all} \quad (9)$$

C1: $a_k \in \{0,1\}$ $\forall_k \in K$

C2: $(1-a_k)\ T_k^l < D_T$ $\forall_k \in K$

C3: $\sum_{k=1}^{K} a_k\ f_k^o < f^{MEC}$ $\forall_k \in K$

C4: $0 < f_k^o < f^{MEC}$ $\forall_k \in K$

Constraints C1 indicate that each UE can only choose one mode to complete the task. Constraints C2 indicates that either executed by local computing or offloading computing, the time cost should not exceed the maximum tolerable delay. Constraints C3 and C4 describe the total computation resources limitation of MEC. obviously, the objective function of problem (11) is not convex, and it is a NP-hard problem [6] such problems are hard to find the optimal solution. In order to solve this problem, we propose reinforcement learning methods, to find the optimal offloading and resource allocation strategy for each UE

III. Problem Solution

A.Reinforcement learning methods

In order to minimize the sum cost of task offloading for all UEs, we propose an A3C-based computing offloading policy, A3C [7] adopts asynchronous training method to reduce the correlation between data, which avoids the lack of experience playback mechanism, and improves the convergence speed by making full use of the advantages of multithreading. In this section, we firstly define the specific state, action and reward of the agent in detail Then, we introduce the proposed scheme which includes the computing offloading policy based on A3C algorithm.

B. Three key elements for DRL

There are three key elements in the reinforcement learning method, namely state, action, reward, specifically to the system model in this article:

State: The system state includes 2 parts of s = (ec, ac). We define ec as the sum cost of the entire system and the available computational capability ac of the MEC server which can be computed as ac=f^{MEC}-$\sum_{k=1}^{K} f_k^o$

Action: The action consists of 2 parts respectively the offloading decision of kUEs A = [α1, α2, . . . , αk] and the available computational capability f = [f1,f2, . . . , fk]

Reward: After executing each possible action a in each step, the agent will get a reward R (s, a) in a certain state The agent can estimate the expected reward for each state-action pair and choose the action that maximizes the reward. the reward function should be related to the objective function. Our objective is to meet the minimal sum cost and the goal of RL is get the maximum reward so the value of reward should be negatively correlated to the size of the sum cost. In this system the immediate reward can be defined as $\frac{eclocal - ec}{eclocal}$

Where $eclocal$ is the sum cost of all tasks executed by local computing and ec gives the actual sum cost of current state

C. A3C Algorithm

At each time slot t, the environment is in state s_t and the estimated state value is V(s_t;θv). The agent executes an action at according to policy π($a_t \mid s_t$;θ) under the current state s_t, and the environment will transfer to a following state s_t+1 under certain probabilities, and the agent will receive a reward r_t. The state value function of A3C is given by

$$V(s_t;\theta v) = E\,[Rt \mid s = s_t, \pi] = E[\textstyle\sum_{k=0}^{\infty} \gamma^k\, r_{t+k} \mid s = s_t, \pi]$$

Then, the advantage function is defined as:

$$A\,(s_t,\ a_t;\theta,\theta v) = Rt - V(s_t;\theta v),$$

On the basis of advantage function At, the loss function of the actor is given by

$$f\pi(\theta) = \log\pi(a_t|s_t;\theta)(Rt - V(s_t;\theta v)) + \beta H(\pi(s_t;\theta))$$

where H(π(s_t;θ)) is an entropy item used for encouraging exploration in training procedure and thus to avoid possible premature convergence, and β is a parameter used to control the strength of the entropy regularization and thus to facilitate the tradeoff between exploration and exploitation.

On the other hand, the loss function associated with the outputs of the critic network is given by

$$f_v(\theta_v) = (R_t - V(s_t;\theta_v))^2$$

Therefore, the iterative formulas of Actor and Critic are respectively:

$$d\theta \leftarrow d\theta + \nabla\theta' \log\pi(a_t|s_t;\theta')(R_t - V(s_t;\theta_v)) + \delta\nabla\theta' H(\pi(s_t;\theta'))$$

$$d\theta_v \leftarrow d\theta_v + \frac{\partial(R_t - V(s_t;\theta_v))^2}{\partial\theta'_v}$$

IV. Simulation Results

In this section, we present the simulation results to evaluate the performance of the proposed scheme. We consider that in a wireless network scenario with a network bandwidth of $W = 10MHz$, each mobile device is randomly distributed within the coverage area. The UEs are randomly scattered within 200m distance away from the eNB. The computation capacity of the MEC server is f^{MEC}=5GHz/s, the maximum CPU frequency of mobile devices f=1GHz/s, and the power during transmission and waiting are respectively $p_k = 500mw$, p_k^i=100mw. [8] And we assume that the data size of the computation offloading B_k (in kbits) obeys uniform distribution between (300, 500), the number of CPU cycles C_k (in Megacycles) obeys uniform distribution between (900,1100). The decision weight of each UE is set to be I_k^t=I_k^e=0.5.

This article will obtain the results through simulation experiments, and reflect the weighted sum of energy consumption and delay in algorithms and strategies such as Full Local, Full Offload, A3C, DQN and Double Q-learning

We first present the sum cost of the MEC system with an increasing total number of UEs in Fig. 2 when the number of users continues to increase, the total cost of all methods is increasing, but under the same conditions, the total cost of the A3C method system is the smallest, then the DQN and Double Q-learning follows with a small gap, the performance of these three methods are relatively stable.

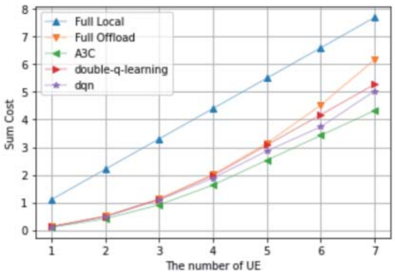

Fig. 2, Sum cost versus the number of UE

As shown in Fig.3, the number of UE is 10. The A3C method gives the best results, the sum costs of all methods increase with the increasing data size of offloading task, because bigger data size leads to more time and energy consumptions for offloading

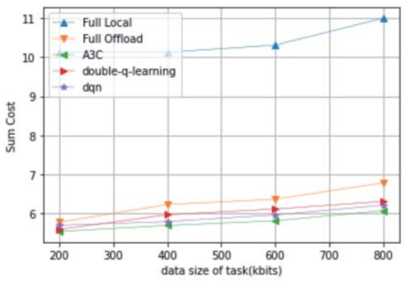

Fig. 3, Sum cost versus the data size of task

As shown in Figure 4, the number of UE is 5. The A3C method gives the best results when the computing power of the MEC server is too small, the total cost of as the computing power of the MEC increases, the total cost of completely offloading the system drops rapidly and the performance of these offloading methods is almost the same. This shows that when MEC has enough computing power, the problem of resource allocation will be weakened, and the superiority of DRL method will no longer be prominent.

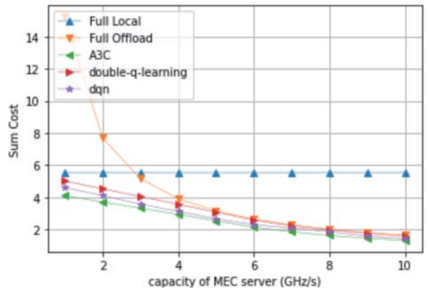

Fig4. Sum cost versus the capacity of the MEC serve

V. Conclusion

In this paper, we presented an integrated framework for multi-user computation offloading and resource allocation with mobile edge computing. From the results of the above simulation experiments, the A3C method, has a better performance in the computational offloading system than the methods. It also shows that increasing enhancing the computing power of edge servers can effectively reduce the total cost of the system and the total cost of the system increases as the amount of task data increases.

References

[1] Jiao L, Friedman R, Fu X, et al. Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities[C]2013 Future Network & Mobile Summit. IEEE, 2013: 1-11. Podani, J. (1994) Multivariate Data Analysis in Ecology and Systematics. SPB Publishing, The Hague.

[2] Zhang J, Letaief K B. (2019) Mobile edge intelligence and computing for the internet of vehicles[J]. Proceedings of the IEEE, 108(2): 246-261

[3] MAO Y, ZHANG J, LETAIEF K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605

[4] N. Kiran, C. Pan, S. Wang and C. Yin, "Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks," in Journal of Communications and Networks, vol. 22, no. 1, pp. 1-11, Feb. 2020, doi: 10.1109/JCN.2019.000046.

[5] J. Li, H. Gao, T. Lv and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," 2018 IEEE Wireless Communications and Networking Conference (WCNC), 2018, pp. 1-6, doi: 10.1109/WCNC.2018.8377343.

[6] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in 2012 Proceedings IEEE INFOCOM, March 2012, pp. 2716–2720.

[7] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1928-1937.

[8] Y. Cao, T. Jiang, and C. Wang, "Optimal radio resource allocation for mobile task offloading in cellular networks," IEEE Network, vol. 28, no. 5, pp. 68–73, September 2014.