



# Learning for Smart Edge: Cognitive Learning-Based Computation Offloading

Yixue Hao<sup>1</sup> · Yingting Jiang<sup>1</sup> · M. Shamim Hossain<sup>2</sup> · Mohammed F. Alhamid<sup>2</sup> · Syed Umar Amin<sup>3</sup>

Published online: 26 September 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

With the development of intelligent applications, more and more intelligent applications are computation intensive, data intensive and delay sensitive. Compared with traditional cloud computing, edge computing can reduce communication delay by offloading computing tasks to edge cloud. Furthermore, with the complexity of computing scenarios in edge cloud, deep learning based on computation offloading scheme has attracted wide attention. However, all the learning-based offloading scheme does not consider the where and how to run the offloading scheme itself. Thus, in this paper, we consider the problem of running the learning-based computation offloading scheme for the first time and propose the learning for smart edge architecture. Then, we give the computation offloading optimization problem of mobile devices under multi-user and multi edge cloud scenarios. Furthermore, we propose cognitive learning-based computation offloading (CLCO) scheme for this problem. Finally, experimental results show that compared with other offloading schemes, the CLCO scheme has lower task duration and energy consumption.

**Keywords** Computation offloading · Cognitive learning · Edge computing · Communication

## 1 Introduction

With the rapid development of wireless technology and cloud computing intelligent devices (such as smartphones [1]) access to the wireless network [2, 3] increasing. Furthermore, mobile devices will be more and more intelligent and the applications in mobile devices will require extensive computation power and persistent data processing [4–6]. However, the improvement of these emerging applications is Limited by the computational power of mobile devices [7,

8]. The computation-intensive task can be offloaded to the cloud for execution by the cloud computing technology, so as to compensate for the restrictions of inadequate computing capability in mobile terminals. However, the mobile devices accessing to the cloud through wireless network needs longer duration, so it is not applicable for the delay sensitive task. These tasks include wearable virtual reality (VR) etc [9].

The edge computing can provide the distributed computing and storage capabilities for mobile devices by deploying the server on the network edge [10], and accordingly can support computing and storing the intensive intelligent application and ensure the lower delay and higher performance [11]. Based on the edge computing, network-based information processing for distributed applications, such as [12], can be realized. Thus, the edge computing has the advantages of low delay, and is an important research direction of wireless communication in the future [13, 14]. Specially, there are two advantages to offload task to edge cloud [15]. On the one hand, compared with mobile devices, edge cloud has more computational resources [16–18]. On the other hand, it can overcome larger delay caused by offloading the computation-intensive task and data-intensive task to cloud [19–21]. Thus, for delay-sensitive and computation-intensive tasks, offloading to the edge cloud may achieve a better tradeoff among delay and energy efficiency [22].

---

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia, for funding this research group No. (RG-1437-042).

---

✉ M. Shamim Hossain  
mshossain@ksu.edu.sa  
Yixue Hao  
yixuehao@hust.edu.cn

- <sup>1</sup> School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
- <sup>2</sup> Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
- <sup>3</sup> Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

For the task offloading based on edge cloud, there exist the following three types of offloading scenes. (i) One user and one server scene, i.e., one mobile user offloading computation task to an edge cloud. (ii) Multi-users and one server scene, i.e., many mobile users offloading the computing task to an edge cloud. (iii) Multi-users and multi-servers scene, i.e., many mobile devices offloading the computing task to many edge clouds servers. In view of different scenes, the researchers design different offloading schemes. Moreover, the complexity of multi-users and multi-servers scene and dense deployment of 5G network (i.e., 5G ultra-dense cellular network) result in the diversification of edge cloud deployment [23, 24]. Thus, the existing optimization-based task offloading scheme is difficult to meet the optimal solution. Fortunately, with the development of artificial intelligence (AI) and especially the development of deep learning, it makes breakthroughs in the field of computer vision [25]. The researchers hope to combine the AI related algorithm with the computation offloading, and enable the edge computation offloading to be more intelligent. For example, Chen et al [26] research the task offloading program by Markov decision process in traditional machine learning algorithm. The test result shows that, such task offloading program has lower delay.

Furthermore, few scholars make task offloading scheme by using deep learning algorithm. In view of computation offloading problem under user’s mobility, Sun et al., put forward the computation offloading scheme of minimizing task duration based on multi-armed bandit theory [27]. Chen et al.,research the edge cloud task offloading program by deep reinforcement learning. However, all above task offloading schemes are running on the mobile device using deep learning algorithm, and fail to consider the following two limitations. (i) The operation of task offloading scheme

based on deep learning needs a lot of computing resources of the mobile devices, i.e., the deep learning algorithm is equivalent to the computing task. The existing task offloading schemes pays close attention to designing the delay and energy efficient algorithm for task offloading. It ignore the computing problem of offloading algorithm. (ii) The computing capability of the mobile device is limited. The offloading scheme based on deep learning may not be able to run, resulting in the failure of the optimal task offloading scheme for the design. Therefore, where to run the learning-based offloading scheme and how to offload the task by deep learning is still a challenging problem.

To solve the above challenges, we focus on the computing problem of task offloading algorithm, and propose the architecture of learning for smart edge. To be specific, we first propose the smart layered based task offloading architecture. Then we introduce the optimal task offloading problem of the mobile device under multi-user and multi-server. Furthermore, we provide the cognitive learning-based computation offloading (CLCO) scheme. Finally, the simulation experiment shows that, the task offloading strategy proposed by us minimizes the task processing delay, and enables the battery energy consumption to be lower. The problem of this paper as shown in Fig. 1.

The main contributions of this paper are summarized as follows:

- We focus on the computing problem of task offloading algorithm, and propose the architecture of learning for smart edge. From two perspectives of computing task offloading scheme learning and task offloading scheme, we give where and how to run the learning-based task offloading scheme.

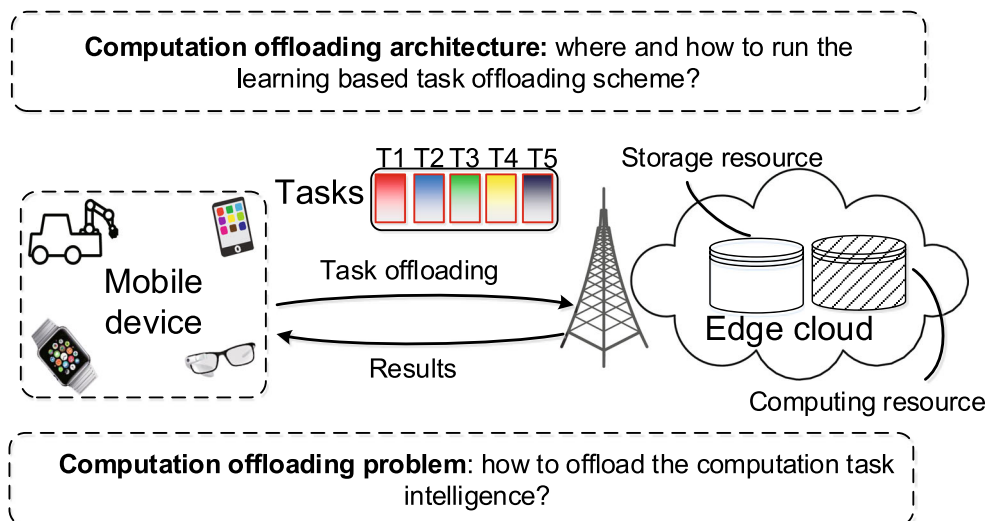


Fig. 1 An illustration of computation offloading in edge cloud

- In view of the specific task offloading problem, based on cognitive learning, we give the optimal task offloading scheme of the mobile device. The simulation experiment result shows that, the CLCO scheme proposed by us has lower task delay and energy consumption.

## 2 System architecture

In this section, we introduce the learning for smart edge architecture from the perspectives of task offloading scheme learning and task offloading scheme. This architecture generally aims at two applications as follows, i.e., (i) computation-intensive application, such as VR and AR, (ii) data-intensive application, such as personalized video service [28–30]. Moreover, these applications are generally delay sensitive [31], and need real-time task offloading. This architecture can make the utmost of computing, storage and network resources of edge server, introduce the cognitive learning method to computation offloading, and process the task offloading scheme learning and task offloading scheme both on the edge cloud, to reduce the computing task delay and improve the quality of user service.

To be specific, we divide the learning for smart edge architecture into three layers, i.e., edge resource cognitive layer, computation task cognitive layer, and global management cognitive layer. The edge resource cognitive layer includes physical resource cognitive layer [32] and virtual resource cognitive layer. It first conducts the software definition on physical resource, data link resource and network resource by the network function virtualization technology [33]. In other words, the physical computing, storage and communication resources are virtualized to form the virtual resources. Then, the software defined virtual resources are encapsulated, to form the edge-centered resource. The computation task cognitive layer includes the offloading task and mobile device cognition. Thus, it can cognizes the required computation amount and transmission for the task, as well as the power consumption of mobile device. Different users generate different tasks, and have different requirements of task delay. The global management cognitive layer is responsible for task offloading learning and the corresponding offloading strategy formulation. Specifically, it formulates the corresponding task offloading strategy by analyzing the offloading task feature and resource state of edge clouds, offloads the task from the global perspective, and accordingly reaches the optimal decision.

For the above mentioned architecture, the specific learning based task offloading flow is shown as follows.

- **Where to run the learning based task offloading scheme.** In this paper, we assume the learning based task offloading schemes are running on the edge cloud

rather than mobile device. It can gain the task offloading strategy on edge cloud based on deep learning algorithm and existing task offloading data training.

- **How to run the learning based task offloading scheme.** When the mobile user generates the task, the mobile device firstly offloads the parameters. The edge cloud learns the optimal task offloading scheme in accordance with its own computing resources. After the edge cloud determines the offloading scheme, it is transmitted to the mobile device. The mobile device offloads accordingly.

## 3 System model

We consider an edge cloud enabled network consisting  $M$  edge clouds. Indexed by  $\mathcal{M} = \{1, 2, \dots, M\}$ , Let  $\mathcal{N} = \{1, 2, \dots, N\}$  denote the set of mobile devices. In this paper, we consider that each mobile device can be connect to multiple edge clouds through wireless channel, and can offload the task to its connected edge cloud. Denote  $\mathcal{A}_i$  as the set of edge clouds that provides computation offloading services to the mobile device  $i$ . We also define  $f_j$  and  $c_j$  as the maximum computation amount and storage capacity of edge cloud  $j$ .

### 3.1 Task model

We assume the computing task to be processed by the mobile device  $i$  as  $Q_i$ . Thereinto,  $Q_i = \{\omega_i, s_i\}$ , where  $\omega_i$  is the computational demand of the task  $Q_i$ , i.e., the CPU cycles and  $s_i$  is the size of computation task  $Q_i$ , i.e., the input size of data contents. For example, as for the emotion recognition task,  $\omega_i$  is the required computing resource for emotion recognition algorithm (such as deep learning algorithm), and  $s_i$  is the emotion data size. In this paper, we assume that the task is separable, i.e. a part of each task can be executed locally, and the other part can be executed by offloading to edge cloud. We denote  $u_{i,j}$  as the ratio of computation amount offloaded by the user  $i$  to the edge cloud  $j$  for the total amount of computation, where  $j \in \mathcal{A}_i$ .

### 3.2 Communication model

We provide the communication model between mobile device and edge cloud. Let  $p_i$  denote transmission power of mobile device  $i$ . Let  $h_{i,j}$  denote channel power gain of mobile device  $i$  and edge cloud  $j$ . Thus, the offloading rate for  $Q_i$  are defined as:

$$r_{i,j} = B \log_2 \left( 1 + \frac{p_i h_{i,j}}{\sigma^2 + I_{i,j}} \right) \quad (1)$$

where  $\sigma^2$  is noise power,  $B$  is channel bandwidth, and  $I_{i,j}$  denotes the interference power between edge clouds. Then, the transmission delay of the mobile device  $i$  offloads the task to the edge cloud  $j$  can be defined as:

$$T_{i,j}^{tra} = \frac{u_{i,j}S_i}{r_{i,j}} \tag{2}$$

Furthermore, the energy consumption when the mobile device  $i$  offloads the task  $Q_i$  to the edge cloud  $j$  can be defined as follows:

$$E_{i,j}^{ec} = \frac{p_i u_{i,j} S_i}{r_{i,j}} \tag{3}$$

### 3.3 Computation model

#### 3.3.1 Local computation

For the computing task that runs locally, we define  $f_i^l$  as the local computing capability in cycle/second of mobile device  $i$ . Thus, we can obtain the execution time of computing task  $Q_i$  locally as follows:

$$T_i^l = \frac{(1 - \sum_{j \in \mathcal{A}_i} u_{i,j}) \omega_i}{f_i^l} \tag{4}$$

Also, the energy consumption of mobile device is defined as follows:

$$E_i^l = \kappa (f_i^l)^2 \left( 1 - \sum_{j \in \mathcal{A}_i} u_{i,j} \right) \omega_i \tag{5}$$

where  $\kappa$  is a constant related to the chip architecture. In this paper, we set  $\kappa = 10^{-25}$ .

#### 3.3.2 Edge cloud computation

We define  $f_{j,i}^{ec}$  as CPU computing capability distributed to the mobile device  $i$  by edge cloud  $j$ . Like in [14] the task duration for task  $Q_i$  executed on edge cloud  $j$  can be defined as:

$$T_{i,j}^{ec} = T_{i,j}^{tra} + T_{i,j}^{proc} = \frac{u_{i,j}S_i}{r_{i,j}} + \frac{u_{i,j}\omega_i}{f_{j,i}^{ec}} \tag{6}$$

Furthermore, the task duration when the task of mobile device  $i$  is executed on edge cloud as follows:

$$T_i^{ec} = \sum_{j \in \mathcal{A}_i} (T_{i,j}^{tra} + T_{i,j}^{proc}) = \sum_{j \in \mathcal{A}_i} \left( \frac{u_{i,j}S_i}{r_{i,j}} + \frac{u_{i,j}\omega_i}{f_{j,i}^{ec}} \right) \tag{7}$$

In a similar way, we can obtain the energy consumption when mobile device  $i$  offload the task on edge cloud as follows:

$$E_i^{ec} = \sum_{j \in \mathcal{A}_i} E_{i,j}^{ec} = \sum_{j \in \mathcal{A}_i} \frac{p_i u_{i,j} S_i}{r_{i,j}} \tag{8}$$

### 3.4 Problem formulation

According to above analysis, we can obtain the task duration and energy consumption of the processing of mobile device  $i$  task. Suppose that the mobile device  $i$  residual energy is  $E_i$ . We are aimed at minimizing the task duration of all tasks with the limited battery capacity. Our optimization variable is  $u_{i,j}$ . The optimization problem can be defined as follow:

$$\begin{aligned} & \underset{u_{i,j}}{\text{minimize}} && \sum_{i=1}^N (T_i^l + T_i^{ec}) \\ & \text{subject to} && C1 : E_i^{ec} + E_i^l \leq E_i \\ & && C2 : \sum_{i=1}^N s_i u_{i,j} \leq c_j, j = 1, 2, \dots, M \\ & && C3 : \sum_{i=1}^N f_{j,i} \leq f_j, j = 1, 2, \dots, M \\ & && C4 : u_{i,j} \in [0, 1] \end{aligned} \tag{9}$$

where the objective function is minimizing the total duration task duration. The first constraint (C1) shows that the energy consumption of task offloading of mobile device  $i$  can not be more than the residual energy consumption. Constraint (C2) ensures the task offloading to the edge cloud can not be more than total edge cloud storage capacity. The third constraint (C3) indicates the amount of computation offloaded to the edge cloud can not exceed the computing capability of edge cloud. Constraint (C4) indicate that the tasks are separable.

## 4 Cognitive learning-based computation offloading

In this section, based on the system model, we introduce the CLCO scheme. In view of the optimization problem (9), the general solution of this problem is utilizing the optimization theory (e.g., convex optimization). However, solving by the optimization theory is based on the following two assumed conditions. (i) From the perspective of mobile device, the mobile device knows the state of edge cloud or assumed to comply with some distribution. (ii) It is assumed that the mobile device has the sufficient computing capability to compute and obtain the optimal offloading

scheme. However, in the 5G scene, such as in 5G ultra-dense cellular network, each small cell is provided with edge cloud [34]. Each mobile device may connect with multiple edge clouds in the vicinity, and the resource state of edge cloud is changed rapidly. Thus, these assumptions are unrealistic.

Thus, we propose the CLCO scheme. The specific idea is as follows. When the mobile device is idle, the pre-computation offloading is conducted in advance, and the pre-computation task offloading strategy is provided. When the task is reached indeed, the reinforcement learning is utilized to further optimize the problem. Suppose the CLCO iteration for  $T$  times. The basic steps are as follows.

- We assume all mobile devices offload task to its connected edge cloud.
- In  $t \in T$  iteration, the mobile device updates its offloading strategy according to the state of edge cloud in the previous iteration.
- In  $t \in T$  iteration, the edge cloud updates its state according to the offloading strategy of mobile device.
- After iteration for  $T$  times, the mobile device provides the optimal offloading strategy.

In the next, we give the CLCO scheme in detail. Suppose interaction  $t \in \{1, 2, \dots, T\}$ . According to above discussion, we define the state space of task offloading as  $s_t = E_t$ , where  $E_t$  is the energy of mobile device in  $t$  iteration. Thus, we can obtain:

$$s_{t+1} = s_t - (E_t^l(t) + E_t^{ec}(t)) \tag{10}$$

where  $E_t^l(t)$  is the energy consumption of mobile device  $i$  in local processing at iteration  $t$ .  $E_t^{ec}(t)$  is the energy consumption of mobile device when the task is processed in edge cloud at iteration  $t$ . Meanwhile we define the action space as:

$$a_t = \left( \left( 1 - \sum_{j \in \mathcal{A}_i} u_{i,j}(t) \right), \sum_{j \in \mathcal{A}_i} u_{i,j}(t) \right) \tag{11}$$

where  $(1 - \sum_{j \in \mathcal{A}_i} u_{i,j}(t))$  is processing locally, and  $\sum_{j \in \mathcal{A}_i} u_{i,j}(t)$  is processing in the edge cloud. Next we will give how to provide the offloading strategy by Q-learning.

When mobile device offload the task, it do not clear the resource state and computation load of edge cloud, so the state of edge cloud should be explored. In this paper, we adopt the model-free Q-learning for the dynamic strategic choice. Specifically, we define  $Q(s_t, a_t)$  as Q-function, showing the value of performing the action  $a_t$  when the

system state is  $s_t$ . For the choice of the action  $a_t$ , we adopt  $\epsilon$ -greedy search strategy, and choose a action among all possible actions uniformly and randomly at the probability of  $\epsilon$  for exploration. Utilize the known best action (i.e., maximizing the Q-function) at the probability of  $1 - \epsilon$ . We define the reward of performing the action as follows:

$$R_t = \beta_l \frac{T_i^{ec}(t) - T_i^l(t)}{T_i^l(t)} + \beta_e \frac{E_i^{ec}(t) - E_i^l(t)}{E_i^l(t)} \tag{12}$$

where  $\beta_l$  and  $\beta_e$  is the weight of mobile device for energy consumption and task duration. In this paper, we adopt Bellman equation to update Q-function, with the details as follows.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta \left( R_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right) \tag{13}$$

To sum up, we provide the specific algorithm as Algorithm 1.

---

**Algorithm 1** Cognitive learning-based task offloading algorithm

---

```

Initialization  $Q(s, a)$ 
Repeat
  Initial state  $S$ 
  Repeat
    Use  $\epsilon$ -greedy policy, and select an action  $a_t$  for execution based on the state  $s_t$ 
    After executing the action, observe reward and new state  $S_{t+1}$ 
     $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta(R_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$ 
     $s_t \leftarrow s_{t+1}$ 
  End

```

---

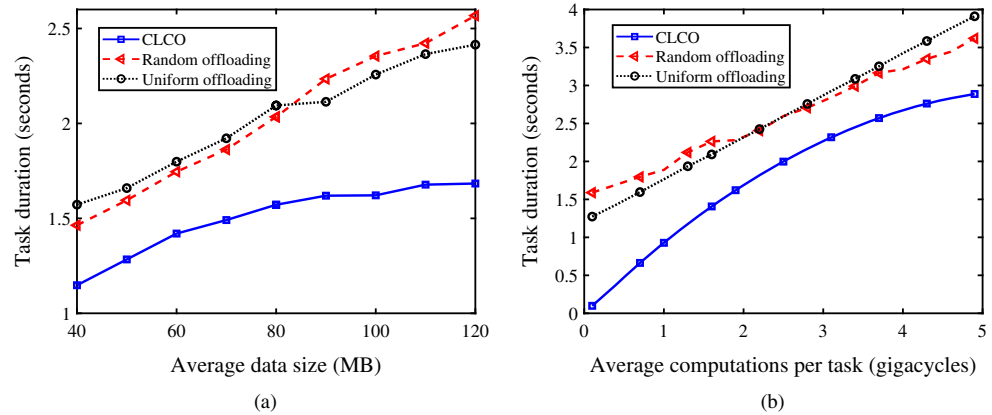
### 5 Performance analysis

In this section, we consider the system involving 5 edge clouds and 300 mobile devices. We set the transmission bandwidth  $B$  of mobile device is 1 MHz, and the transmitting power  $P$  is 0.2 W. The corresponding noise power  $\sigma^2$  and channel power gain  $h$  are  $10^{-9}$  W and  $10^{-5}$ . We assume computation amount  $\omega_i$  and data size  $s_i$  follow by a probability distribution. Specifically the task computation demand and the input data size comply with the normal distribution. The computing capability of edge cloud and mobile device is 10 GHz and 1GHz, respectively.

We first study the CLCO scheme and compared the following computation offloading strategies. (i) Random



**Fig. 2** Task duration achieved by random offloading, uniform offloading and CLCO for various value of (a) the average data size of per task and (b) the average computations capacity of per task



offloading scheme: mobile device randomly offload the computing task, till meeting the computing capacity of edge cloud. (ii) Uniform offloading scheme: The mobile device offloading the computation task uniform on the edge cloud, till reaching the computation capacity of edge cloud. Figure 2 shows the comparison of task duration among random offloading, uniform offloading and CLCO scheme under different data size and computation capacity of per task. We observe that the CLCO scheme outperform the random offloading and uniform offloading strategy when the average data size and average computations per task is increased. This can be explained by the fact that both the random offloading and uniform offloading scheme does not recognize the task.

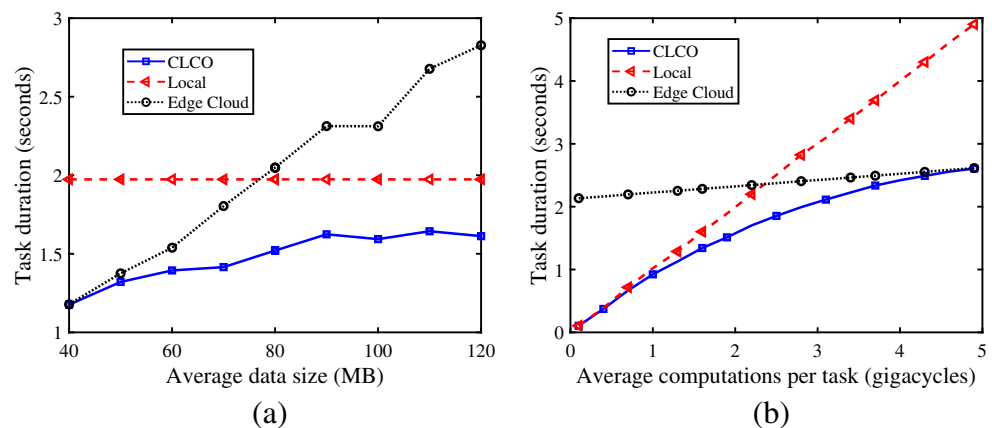
Furthermore, we compare the CLCO strategy with the local computing and edge computing strategy. Figure 3 shows the comparison of task duration among local computation, edge cloud computation and CLCO scheme under different data size and computation capacity of per task. We observe that the CLCO scheme outperform the

local computation and edge cloud computation when the average data size and average computations per task is increased.

### 6 Conclusion

In this paper, we first introduce a new task offloading architecture which include computation task cognitive layer, edge resource cognitive layer and global management cognitive layer. Then, we give the optimization problem of computation offloading under multi-user and multi-server scene. Furthermore, we propose the CLCO scheme. It study about the computing problem of the learning-based computation offloading algorithm. Experiments indicate that the CLCO scheme outperforms several state-of-the-art offloading scheme. In the future, we will do deeper work about the optimization of learning-based computation offloading scheme, and to further reduce the task duration and energy consumption.

**Fig. 3** Task duration achieved by local computing, edge computing and CLCO for various value of (a) the average data size of per task and (b) the average computations capacity of per task



## References

1. Liu H, Zhang B, Su X, Ma J, Wang W, Leung K (2017) Energy-aware Participant Selection for Smartphone-enabled Mobile Crowd Sensing. *IEEE Syst J* 11(3):1435–1446
2. Lu H, Li Y, Chen M, Kim H, Serikawa S (2018) Brain intelligence: go beyond artificial intelligence. *Mob Netw Appl* 23(2):368–375
3. Hao Y, Peng L, Hu H, Hassan M, Atif A (2017) Energy harvesting based body area networks for smart health. *Sensors* 17(7):1602
4. Zhang Y, Gravina R, Lu H, Massimo V, Giancarlo F (2018) PEA: Parallel Electrocardiogram-based authentication for smart healthcare systems. *J Netw Comput Appl* 117:10–16
5. Liu H, Zhao J, Zhang H, Guo S, Leung K, Crowcroft J (2017) Energy-Efficient Event detection by participatory sensing under budget constraints. *IEEE Syst J* 11(4):2490–2501
6. Liu H, Fan J, Hui P, Wu J, Leung K (2015) Towards QoI and Energy-Efficiency in Participatory Crowdsourcing. *IEEE Trans Veh Technol* 64(10):4684–4700
7. Ge X, Yang B, Ye J, Mao G, Wang C, Han T (2015) Spatial spectrum and energy efficiency of random cellular networks. *IEEE Trans Commun* 63(3):1019–1030
8. Jiang H, Cai C, Ma X, Yang Y, Liu Q (2018) Smart home based on WiFi sensing: a survey. *IEEE Access* 6(6):13317–13325
9. Ge X, Pan L, Li Q, Mao G, Tu S (2017) Multi-Path Cooperative communications networks for augmented and virtual reality transmission. *IEEE Trans Multimed* 19(10):2345–2358
10. Yu N, Kaoru S (2018) Per-Flow Throughput fairness in ring aggregation network with multiple edge routers. *Big Data Cogn Comput* 2(3):17
11. Chen M, Hao Y, Hu L, Hossain M, Ghoneim A (2018) An adaptive fusion strategy for distributed information estimation over cooperative Multi-Agent networks. *IEEE Trans Inf Theory* 63(5):3076–3091
12. Tian D, Zhou J, Sheng Z (2017) Edge-cocaco: Towards Joint Optimization of Computation, Caching and Communication on Edge Cloud. *IEEE Wirel Commun* 25(3):21–27
13. Tong L, Li Y, Gao W (2016) A hierarchical edge cloud architecture for mobile computing. *Proceedings of the 35th IEEE Conference on Computer Communications*, pp 399–400
14. Chen X, Jiao L, Li W, Fu X (2016) Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking* 24(5):2795–2808
15. Hany F, Robert J, Gary B (2018) Fog Computing and the Internet of Things: A Review. *Big Data Cogn Comput* 2(2):10
16. Chen M, Hao Y, Hu L, Huang K, Lau NK (2017) Green and Mobility-aware Caching in 5G Networks. *IEEE Trans Wirel Commun* 16(12):8347–8361
17. Wang C, Li Y, Jin D (2014) Mobility-Assisted Opportunistic computation offloading. *IEEE Commun Lett* 18(10):1779–1782
18. Chen M, Hao Y, Li Y, Lai C, Wu D (2015) On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Commun Mag* 53(6):18–24
19. Chun B, Ihm S, Maniatis P, Naik M, Patti A (2011) Clonecloud: elastic execution between mobile device and cloud. *Proceedings of the sixth conference on Computer systems*, pp 301–314
20. Barbera M, Kosta S, Mei A, Stefa J (2013) To offload or not to offload the bandwidth and energy costs of mobile cloud computing. *INFOCOM, 2013 Proceedings IEEE*, pp 1285–1293
21. Kollenstart M, Harmsma E, Langius E, Andrikopo V, Lazovik A (2018) Adaptive provisioning of heterogeneous cloud resources for big data processing. *Big Data Cogn Comput* 2(3):15
22. Chen M, Hao Y, Qiu M, Song J, Wu D, Humar I (2016) Mobility-aware Caching and Computation Offloading in 5G Ultradense Cellular Networks. *Sensors* 16(7):974–987
23. Li Y, Wang W (2014) Can mobile cloudlets support mobile applications?. In: *2014 Proceedings IEEE INFOCOM*, pp 1060–1068
24. Chen M, Qian Y, Hao Y, Song L (2018) Data-Driven Computing and Caching in 5G Networks: Architecture and Delay Analysis. *IEEE Wirel Commun* 25(1):70–75
25. Joel RB (2018) A deep learning model of perception in Color-Letter synesthesia. *Big Data Cogn Comput* 2(1):8
26. Chen M, Hao Y (2018) Task Offloading for Mobile Edge Computing in Software Defined Ultra-dense Network. *IEEE J Sel Areas Commun* 36(3):587–597
27. Sun Y, Zhou S, Xu J (2017) EMM: Energy-Aware mobility management for mobile edge computing in ultra dense networks. *IEEE J Sel Areas Commun* 35(11):2637–2646
28. Daniel B (2018) Reimaging research methodology as data science. *Big Data Cogn Comput* 2(1):4
29. Zheng Y, Wu D, Ke Y, Yang C, Chen M, Zhang G (2017) EMM: Energy-Aware mobility management for mobile edge computing in ultra dense networks. *ZhangYuanhuan* 27(8):1777–1789
30. Wang D, Peng Y, Ma X, Ding W, Jiang H, Liu J (2018) Adaptive Wireless Video Streaming based on Edge Computing: Opportunities and Approaches. *IEEE Transactions on Services Computing*. <https://doi.org/10.1109/TSC.2018.2828426>
31. Chen X, Pu L, Gao L, Wu W, Wu D (2017) Exploiting massive D2D collaboration for Energy-Efficient mobile edge computing. *IEEE Wirel Commun* 24(4):64–71
32. Kang K, Chen L, Yi H, Wang B, Sha M (2017) Real-Time Information derivation from big sensor data via edge computing. *Big Data Cogn Comput* 1(1):5
33. Chen M, Miao Y, Hao Y, Hwang K (2017) Narrow band internet of things. *IEEE Access* 5:20557–20577
34. Ge X, Sun Y, Gharavi H, John T (2018) Joint Optimization of Computation and Communication Power in Multi-user Massive MIMO Systems. *IEEE Trans Wirel Commun* 17(6):4051–4063