Service-dependent task offloading for multiuser mobile edge computing system

Wanli Ni, Hui Tian✉, Xinchen Lyu and Shaoshuai Fan

Mobile edge computing (MEC) has been deemed as one of the key technologies for pushing powerful computing ability to the radio access network. Different scenarios require different services, and tasks usually require computing in a specific runtime environment. However, few studies have taken this into consideration. To this end, the authors propose a novel model of service-dependent task offloading for the multiuser MEC system with resource constraints. Through determining which services should be deployed at the network edge and how many tasks should be offloaded, a revenue maximisation problem is constructed. Accordingly, they propose an iterative optimisation algorithm with high performance for the service deploying and task offloading problem. Finally, numerical simulations reveal that their proposed algorithm outperforms other schemes.

Introduction: By enabling cloud-computing capabilities at the network edge, mobile edge computing (MEC) can provide flexible and on-demand services in close proximity to mobile devices. Meanwhile, it makes the ultra-reliable and low-latency processing of the massive real-time data a reality [1]. In the era of information explosion, the services are scaling with the expanding of user demands, and the MEC is becoming increasingly service-dependent. For one thing, the edge servers are storage and computing limited [2], and it would be impossible to have all services deployed on it. Hence, it deserves much attention to decide which services should be deployed at the network edge. For another, the wireless base station (BS) has limited communication resources [3], which may not satisfy all users for data uploading. Thus, it is worth considering how many tasks should be offloaded.

So far, many previous studies [3–5] on resource management and service placement have been conducted separately in the MEC. However, the decisions of resource management and service placement are dependent on each other. Separate optimisation would lead to the performance loss. In the literature, Chen *et al*. [3] investigated a game theory approach for the allocation of wireless channels among multiusers. However, it assumed that all user requirements could be met by the edge server. In fact, hardware resources are finite in practical cases, and the total required computing resources should not exceed what can be provided. The work [4] proposed a semidistributed offloading strategy to maximise the quality of experience (QoE)-based utility via quasi-convex and convex optimisation. Although resource constraints are considered in [4], the software environments for task execution are ignored, and not all services can be supported in the real world. Based on the Lyapunov optimisation, the work [5] presented a mobility-aware online algorithm for service placement. However, it only optimised the location of service deployment and ignoring the dependencies between task offloading and service placement.

Distinctively different from the existing work, we take into account the software runtime environment for task execution and jointly optimise it with task offloading from the perspective of a real case. The joint optimisation enlarges the search regions of the network operations and enforces the mixed-integer feature of the problem, which can bring more revenues to operators and provide a reference for the practical application of MEC in future networks.

System model: In this Letter, we consider a system consisting of J users and one MEC-enabled BS. We denote the set of wireless channels as $\mathcal{I} = \{1, 2, \ldots, I\}$. Let $\mathcal{J} = \{1, 2, \ldots, J\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of users and tasks, respectively. For convenience, k also refers to the service that supports task k. Then, we define $a_k \in \{0, 1\}, \forall k$ as the service deploying decision. Specifically, we have $a_k = 1$ if the service k is deployed on the MEC server, and $a_k = 0$ otherwise.

Owing to service heterogeneity, we assume that each service k requires S_k storage size to install the software environment, and the maximum storage size of the MEC server is $S_{\max}$. Thus, the service deploying should satisfy the storage constraint

$$\sum_{k=1}^{K} a_k S_k \leq S_{\max}. \tag{1}$$

Additionally, we assume that each service k requires V_k virtual machine (VM) units when deploying it on the MEC server. To guarantee the performance of each VM unit, the maximum number of VM units generated by the MEC server should not exceed $V_{\max}$. Thus, the VM constraint can be expressed as follows:

$$\sum\nolimits_{k=1}^{K} a_k V_k \leq V_{\max}. \tag{2}$$

In the long run, users share the channels with each other. Therefore, we consider a time average task offloading scenario of long term, where the allocation proportion of channel capacity between the channel i and user j for task k is denoted as $b_{i,j,k} \in [0, 1], \forall i, j, k$. Only when the service is deployed, task offloading is effective, i.e. when $a_k = 1$, then $b_{i,j,k} \geq 0$. Otherwise, $b_{i,j,k} = 0$. Thus, the following constraint of channel capacity must be held

$$\sum_{j=1}^{J}\sum_{k=1}^{K} b_{i,j,k} \leq 1, \quad \forall i. \tag{3}$$

Owing to the existence of pass loss and fast fading, the power gain among users and channels is different. Thus, the time average aggregate channel capacity between channel i and user j for all tasks can be defined as $R_{i,j}$ bits during a given offloading period. Moreover, when task k is offloaded to the MEC server, e_k CPU cycles per bit are required to execute [6], but the MEC server has finite CPU cycles $C_{\max}$. Thus, the offloading data of all tasks should satisfy the following computation constraint:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} b_{i,j,k} R_{i,j} e_k \leq C_{\max}. \tag{4}$$

Problem formulation: In real cases, the mobile network operator (MNO) will charge users for executing tasks on behalf of them. Hence, our goal is to maximise the total revenue from the perspective of MNO, while meeting the above-mentioned resource constraints. Then, the unit price for the MEC server to compute task k from users is defined as p_k revenues per bit. Accordingly, the problem can be formulated as follows:

$$\begin{aligned} \max_{A,B} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} a_k b_{i,j,k} R_{i,j} p_k \\ \text{s.t.} \quad & (1) - (4), \end{aligned} \tag{5}$$

where $A = \{a_k | \forall k \in \mathcal{K}\}$ and $B = \{b_{i,j,k} | \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}\}$. Since $a_k \in \{0, 1\}$ and $b_{i,j,k} \in [0, 1]$ are binary and continuous variables, respectively, problem (5) is a mixed integer programming (MIP) problem. What is more, judging from partial constraints (1) and (2), we can infer that problem (5) includes an extended version of the knapsack problem, which makes the original problem NP-hard. Thus, it is difficult to obtain the optimal solution of problem (5) in closed form. Fortunately, we note that it could be solved by decomposing the MIP problem into the following two dependent subproblems:

- *Task offloading problem*: When the decision of service deploying is given, i.e. $A = A^{(0)}$, problem (5) becomes a convex optimisation problem about the task offloading strategy B.
- *Service deploying problem*: When the offloading strategy B is fixed, i.e. $B = B^{(0)}$, problem (5) is transformed into a 0-1 programming problem about the service deploying variable A.

Algorithm design: According to the mathematical property of these two subproblems, we can use the branch and bound method (BBM) to obtain the optimal A^*, and the interior point method (IPM) can be used to find the optimal B^*. More details about the iterative optimisation of service deploying and task offloading (IOSDTO) are given in Algorithm 1, and the complexity analysis is as follows.

In Algorithm 1, the complexity of the standard IPM is $\mathcal{O}(N_{\text{ip}} L^3)$, where $L = IJK$ is the dimension of variable B, and N_{ip} is the number of iterations for finding the optimal task offloading strategy. In the worst case, the complexity of the BBM is $\mathcal{O}(N_{\text{bb}} 2^K)$ where K is the dimension of variable A, and N_{bb} is the number of iterations for optimising the service deploying problem. Overall, while the stopping criteria are reached, the number of master iteration can be defined as N_m

where $N_m \leq M$. Therefore, the total complexity of Algorithm 1 is $\mathcal{O}(MN_{\text{ip}}L^3 + MN_{\text{bb}}2^K)$, which is exponential. Fortunately, K is small, and $L \gg K$ in most instances.

Algorithm 1: IOSDTO algorithm for solving problem (5)

1: Initialise $A^{(0)}$, $B^{(0)}$, $R_{i,j}$, p_k, S_k, $S_{\max}$, V_k, $V_{\max}$, e_k, $C_{\max}$, tolerance ξ, iteration number m, and maximum iteration number M.
2: Compute the utility value $U^{(0)} = U(A^{(0)}, B^{(0)})$, where $U(A, B)$ is the objective function of problem (5).
3: With a given $A^{(m)}$, obtain $B^{(m+1)}$ by solving the task offloading problem with the IPM.
4: With a given $B^{(m+1)}$, obtain $A^{(m+1)}$ by solving the service deploying problem with the BBM.
5: Compute $U^{(m+1)} = U(A^{(m+1)}, B^{(m+1)})$. If $|U^{(m+1)} - U^{(m)}| < \xi$ or $m > M$, terminate. Otherwise, set $m = m + 1$ and go to step 3.

Simulation settings and results: In our service-dependent task offloading network composed of one MEC-enabled BS and multiuser, we set the bandwidth of each channel as 500 kHz, and all users can transmit data to the BS at 23 dBm. The pass loss model refers to $127 + 30\log_{10}(d)$ where d (km) is the distance between the BS and users. We assume that the deviation of shadow fading is 4 dB, and the noise power density is -174 dBm/Hz [4]. When it comes to task offloading, the unit price for the MEC server to process data on behalf of users is considered as a constant. Meanwhile, we assume that all of the required storage size S_k, VM units V_k, and CPU cycles e_k follow the uniform distribution [7] (e.g. $S_k \sim U(10, 100)$ GB/service, $V_k \sim U(1, 5)$ units/service, and $e_k \sim U(200, 400)$ cycles/bit). Furthermore, the maximum storage size $S_{\max}$, VM numbers $V_{\max}$ and CPU cycles $C_{\max}$ are 500 GB, 20 units, and 10 gigacycles, respectively, unless otherwise stated. We simulate 2000 runs, and compare Algorithm 1 with the following schemes:

- *Optimal service deploying and equal task offloading (OSDETO)*: All users are allocated the same channel capacity for the task offloading when the service deploying brings the maximal total revenue (e.g. the service deploying problem is solved by the BBM).
- *Random service deploying and optimal task offloading (RSDOTO)*: Services are deployed randomly, and the task offloading strategy is obtained by the IPM.
- *Random service deploying and equal task offloading (RSDETO)*: All users are allocated the same channel capacity, while services are deployed randomly.

Fig. 1 shows the empirical CDF of our proposed algorithm and the comparison schemes. We can see that there are two groups of curves with close performance. One group consists of the OSDETO and RSDETO scheme, and the other group consists of the IOSDTO and RSDOTO scheme. In the latter group, we note that the probability of revenue $> 4 \times 10^7$ is more concentrated, and the curves are closer to the right side than the former group, i.e. the latter group performs better than the former. Thereinto, our proposed IOSDTO algorithm is the champion, and the RSDOTO scheme is the runner-up.

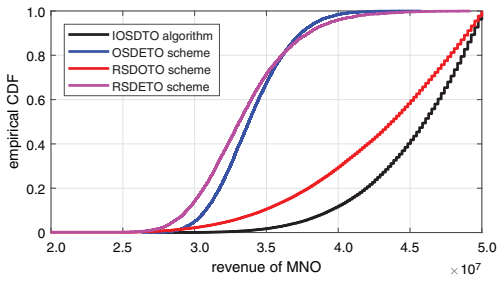

Fig. 1 *Empirical CDF ($I = J = K = 10$)*

Fig. 2 shows the revenue of MNO versus the number of channels. From this figure, it is observed that the total revenue of MNO increases with the number of channels at the beginning. This is due to the fact that as the number of channels increases, more tasks can be offloaded to the edge server over the wireless channels. However, when the edge server cannot provide more CPU cycles for the offloading data, even if the channel capacity is larger, there will not be more revenue. For this reason, we can find that computation resource is in short supply when the number of channels is > 6 under our simulation settings.

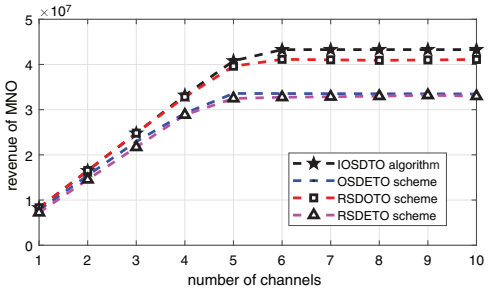

Fig. 2 *Revenue versus channels ($I \in [1, 10]$, and $J = K = 10$)*

Fig. 3 shows the revenue of MNO versus the maximum computation capacity of the edge server. In this figure, we change the maximum computation ability of the edge server, and other parameters are the same as the last point in Fig. 2. As the CPU cycles increase, more tasks can be offloaded from users and executed by the edge server, so the total network revenue grows. With the analysis in the previous paragraph, we have sufficient reasons to believe that the supply and demand relationship of resources is one of the main factors impacting on the total revenue of MNO. Besides, Fig. 3 also reconfirms the effectiveness of Algorithm 1.

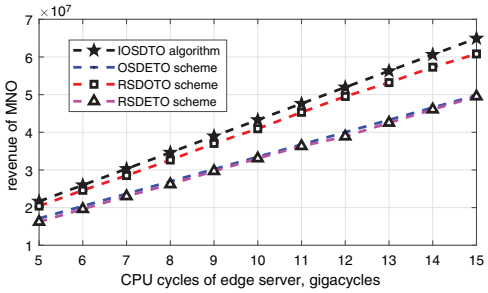

Fig. 3 *Revenue versus cycles ($C_{\max} \in [5, 15]$, and $I = J = K = 10$)*

Conclusion: In this Letter, we first present a novel model of service-dependent task offloading for the multiuser MEC system and investigate the optimal service deploying and task offloading strategy. To maximise the total revenue of MNO, we formulate a MIP problem which is NP-hard. For solving this challenging problem, we decompose it into two simple but dependent subproblems and propose an iterative optimisation algorithm to address them interactively. Eventually, simulation results demonstrate the effectiveness of our proposed algorithm.

Acknowledgments: This Letter was supported by the Beijing Natural Science Foundation (L182036), the Fundamental Research Funds for the Central Universities (2018RC01), and the BUPT Excellent Ph.D. Students Foundation (XTCX201830).

Submitted: *08 April 2019* E-first: *11 June 2019*
doi: 10.1049/el.2019.1179
One or more of the Figures in this Letter are available in colour online.

Wanli Ni, Hui Tian, Xinchen Lyu and Shaoshuai Fan (*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China*)

✉ E-mail: tianhui@bupt.edu.cn

References

1 Bennis, M., Debbah, M., and Poor, H.V.: 'Ultrareliable and low-latency wireless communication: tail, risk, and scale', *Proc. IEEE*, 2018, **106**, (10), pp. 1834–1853

2 Mao, Y., You, C., Zhang, J., *et al*.: 'A survey on mobile edge computing: the communication perspective', *Commun. Surv. Tutors.*, 2017, **19**, (4), pp. 2322–2358

3 Chen, X., Jiao, L., Li, W., *et al*.: 'Efficient multi-user computation offloading for mobile-edge cloud computing', *IEEE/ACM Trans. Netw.*, 2016, **24**, (5), pp. 2795–2808

4 Lyu, X., Tian, H., Zhang, P., *et al*.: 'Multi-user joint task offloading and resources optimization in proximate clouds', *Trans. Veh. Technol.*, 2017, **66**, (4), pp. 3435–3447

5 Ouyang, T., Zhou, Z., and Chen, X.: 'Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing', *J. Sel. Areas Commun.*, 2018, **36**, (10), pp. 2333–2345

6 You, C., Huang, K., Chae, H., *et al*.: 'Energy-efficient resource allocation for mobile-edge computation offloading', *Trans. Wirel. Commun.*, 2017, **16**, (3), pp. 1397–1411

7 Xu, J., Chen, L., and Zhou, P.: 'Joint service caching and task offloading for mobile edge computing in dense networks'. IEEE INFOCOM 2018 – IEEE Conf. Computer and Communication, Honolulu, HI, USA, April 2018, pp. 207–215