Joint Computation Offloading and Resource Allocation for Mobile Edge Computing

John Erskine(✉) and Dang Van Huynh

School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast, UK
{jerskine07,dhuynh01}@qub.ac.uk

Abstract. This paper studies joint computation offloading and communication resource allocation for mobile edge computing (MEC) systems. We aim to minimise the end-to-end (e2e) latency among users (UEs) by jointly optimising task offloading portions, frequency of UEs and edge servers, as well as transmission power of UEs with respect to the latency requirements, computing and energy budgets. To deal with this challenging optimisation problem, we propose efficient algorithms based on the alternating optimisation (AO) approach and convex optimisation. Simulation results validate the effectiveness of the proposed solutions in terms of reducing the e2e latency of UEs as well as demonstrate the impacts of involved parameters on the performance.

Keywords: Mobile Edge Computing · Task Offloading · Resource Allocation · Convex Optimisation

1 Introduction

1.1 Mobile Edge Computing

Mobile Edge Computing (MEC), is a novel technology in the field of mobile computing that allows computing, storage, and networking services to be deployed in nearby edge servers [8,19]. MEC plays an important role in 5G and beyond. Services in terms of computing perspective have strong potential to integrate with other wireless technologies as URLLC [17] and mm-wave [4] which enable new applications. The main objective of MEC is to provide ultra-low latency, high-bandwidth, secure services, and direct access to real-time network information to mobile users (UEs), by bringing the cloud to the edge of the network [17]. MEC is considered a key enabler for emerging mobile applications such as augmented reality, virtual reality, Internet of Things (IoT) and 5G services [3]. MEC provides a cost-effective solution for providing computing services to mobile users by reducing the amount of data that needs to be transmitted to the central cloud.

N.-S. Vo and H.-A. Tran (Eds.): INISCOM 2023, LNICST 531, pp. 51–62, 2023.
https://doi.org/10.1007/978-3-031-47359-3_4

Research in MEC has focused on several areas such as resource management, security, quality of service (QoS), energy efficiency. Resource management algorithms, such as the one in this paper, have been proposed to optimise the use of computational and storage resources at the edge of the network [16]. Quality of service is a critical aspect of MEC, as it ensures that the services provided by the edge server meet the required performance levels [6,12]. Energy efficiency is another important aspect, as it helps to reduce the energy consumption of mobile devices and edge servers [13]. MEC has the potential to revolutionise the way we use mobile devices by bringing the clouds computing power to mobile devices.

1.2 Resource Allocation

One of the key challenges in MEC is resource allocation, which involves deciding how to allocate computational and storage resources at the edge of the network to meet the demands of UEs. Resource allocation strategies in MEC are essential to optimise the use of resources, minimise energy consumption [13], and ensure that the QoS requirements of UEs are met.

Several such resource allocation strategies have been proposed in previous literature to address the challenges of resource allocation in MEC. These include centralised algorithms, distributed algorithms [18], and hybrid algorithms [14]. Centralised algorithms allocate resources based on a global view of the system, whilst decentralised algorithms allocate resources based on local information. Hybrid algorithms can combine the advantages of centralised and decentralised algorithms to provide a balance between performance and scalability. Another important aspect of resource allocation in MEC is the integration of energy efficiency. Several energy-efficient resource allocation algorithms have been proposed to address this challenge [20].

1.3 Task Offloading

Task offloading or computing offloading, involves migrating computing processes from a mobile device to a MEC server. Task offloading is performed to shorten the task response delays and save energy resources [1]. Task offloading in MEC refers to the process of transferring computationally intensive tasks from UEs to edge servers for processing. This is an important aspect of MEC as it helps to reduce the energy consumption of UEs and improves their performance. Several energy-efficient task offloading algorithms have been proposed in literature to address this challenge [20]. More importantly, task offloading in MEC has been recently integrated into other emerging technologies like UAV-based communications and digital twin [2,7] to fully realise the potential of MEC in the development of new delay-sensitive applications.

2 System Model and Problem Formulation

We consider a MEC system as illustrated in Fig. 1. In the system model, the user layer consists of many user equipment (UE) such as mobile robots, actu-

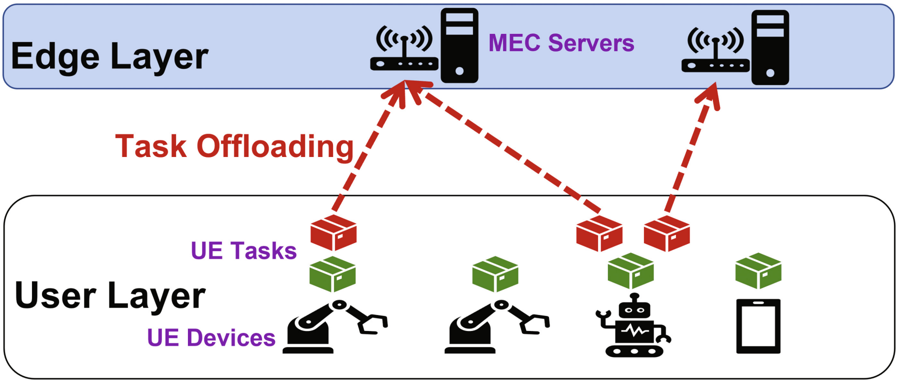

Fig. 1. An illustration of MEC System Model.

ators, sensors. The UEs have different computational tasks to execute, but the computing capacity of UEs is typically limited. Therefore, a portion of tasks can be offloaded to the edge server (MEC) on the edge layer to reduce execution time. Each UE can transmit data to multiple MECs to offload a computational task, and each MEC is available to serve all UEs with an appropriate sharing computing resource to guarantee performance.

2.1 Task Offloading Model

Let us define $\mathcal{M} = \{1, 2, ..., M\}$ and $\mathcal{K} = \{1, 2, ..K\}$ as the sets of UEs, and MECs in the system, respectively. A computational task offloaded from the m-th UE is modelled by three parameters including data size (bit), D_m, the required computation CPU cycles to execute the task, C_m (cycle), and $T_m^{\max}$ (s) is the maximum latency tolerance. Let $\mathbf{u} = \{u_m\}_{\forall m}$ be the portion of the task executed locally and $\mathbf{v} = \{v_{mk}\}_{\forall m,k}$ be the portion of the task offloaded from the m-th UE to the k-th MEC server, satisfying $0 \leq u_{mk} \leq 1$, $0 \leq v_{mk} \leq 1$. Following that definition, the constraint of each computational task can be expressed as follows

$$u_m + \sum_{k \in K} v_{mk} = 1, \forall m. \tag{1}$$

2.2 Wireless Transmission Model

Task offloading from UEs to MECs is based on wireless communications. The access points (AP) associated with MECs are equipped with L antennas to serve single-antenna UEs. The wireless channel between the m-th UE and the k-th MEC is modelled as $\mathbf{h}_{mk} = \sqrt{\gamma_{mk}}\mathbf{g}_{mk}$, where γ_{mk} is the channel path-loss and $\mathbf{g}_{mk}$ is the small-scale fading coefficients. We apply maximum ratio combining (MRC) at the AP to calculate the transmission rate for task offloading and first introduce $\varphi_{m,k,l} = \mathbf{h}_{mk}^T \mathbf{h}_{lk}^* / \|\mathbf{h}_{lk}\|$. Then, the achievable transmission rate from the m-th UE to k-th MEC can be calculated as [9]

$$R_{mk}(\mathbf{p}) = B \log_2 \left(1 + \frac{p_m |\varphi_{m,k,m}|^2}{\mathcal{I}_m(\mathbf{p}) + \sigma_k^2}\right), \tag{2}$$

where B, p_m, and σ_k^2 are the system bandwidth, transmission power of the m-th UE, and the noise variance, respectively. The expression of $\mathcal{I}_m(\mathbf{p}) = \sum_{l \in \mathcal{M}, l \neq m} p_l |\varphi_{m,k,l}|^2$ is the interference caused by other UEs. As a result, the transmission latency for task offloading from the m-th UE to the k-th MEC is expressed as

$$T_{mk}^{\mathsf{off}}(\mathbf{p}, u_m) = \frac{v_{mk} D_m}{R_{mk}(\mathbf{p})}. \tag{3}$$

2.3 Computation and Latency Models

Let f_m^{ue} be the frequency (i.e., clock speed or processing rate) of the m-th UE (cycles/s). The latency of local execution at the m-th UE is given by

$$T_m^{\mathsf{ue}}(u_m, f_m^{\mathsf{ue}}) = \frac{u_m C_m}{f_m^{\mathsf{ue}}} \tag{4}$$

Similarly, given the frequency f_k^{mec}, the latency of the k-th MEC server to execute a portion v_{mk} of the task offloaded from the m-th UE is modelled as

$$T_{mk}^{\mathsf{mec}}(v_{mk}, f_{mk}^{\mathsf{mec}}) = \frac{v_{mk} C_m}{f_{mk}^{\mathsf{mec}}} \tag{5}$$

Consequently, the end-to-end (e2e) latency to complete task execution including local processing, wireless transmission, and MEC processing is expressed as following T_m^{tot}. This value cannot exceed the maximum latency requirement, $T_m^{\max}$

$$\begin{aligned} T_m^{\mathsf{tot}}(\mathbf{p}, u_m, v_{mk}, f_m^{\mathsf{ue}}, f_{mk}^{\mathsf{mec}}) = T_m^{\mathsf{ue}}(u_m, f_m^{\mathsf{ue}}) + \max\{T_{mk}^{\mathsf{off}}(\mathbf{p}, v_{mk})\} \\ + \max\{T_{mk}^{\mathsf{mec}}(v_{mk}, f_{mk}^{\mathsf{mec}})\} \leq T_m^{\max}, \forall m. \end{aligned} \tag{6}$$

It is important to note that after the MEC executes the offloaded task, the size of responses from MECs to UEs (e.g., controlled packets) are much smaller than the offloaded data size, whilst the transmission power of APs are higher than UEs. Therefore, the latency for responses from MECs to UEs is not considered in this paper [19].

2.4 Energy Consumption Model

To execute computational tasks locally, and to offload the task to APs via wireless links, UEs consume energy for computation as well as communication. We model the total energy consumption of UEs as a summation of two components including energy for computation (E_m^{cp}) and energy for communication (E_m^{cm}). As a constraint, this value must be less than or equal to the energy budget of UEs.

$$\begin{aligned} E_m^{\mathsf{tot}}\left(u_m, p_m, f_m^{\mathsf{ue}}\right) &= E_m^{\mathsf{cp}} + E_m^{\mathsf{cm}} \\ &= u_m \frac{\theta}{2} C_m (f_m^{\mathsf{ue}})^2 + \sum_{k \in \mathcal{K}} p_m \frac{v_{mk} D_m}{R_{mk}(\mathbf{p})} \leq E_m^{\max}, \forall m, \end{aligned} \tag{7}$$

where $\theta/2$ is a constant factor to calculate the computation energy consumption of the m-th UE [15].

2.5 Problem Formulation

In this paper, we consider a min-max fairness latency minimisation problem. In particular, we aim at minimising the worst-case e2e latency among UEs subject to requirements of the latency, the UEs energy budget, the transmission power budget, the offloading policies, and the computation capacity of UEs and MECs. On this point, the optimisation problem is formulated as follows

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{p}, \mathbf{f}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{\mathsf{tot}}\left(u_m, v_{mk}, f_m^{\mathsf{ue}}, f_{mk}^{\mathsf{mec}}, \mathbf{p}\right) \right\} \tag{8a}$$

$$\text{s.t. } p_m \leq P_m^{\max}, \forall m, \tag{8b}$$

$$u_m \in [0,1], v_{mk} \in [0,1], \forall m, k, \tag{8c}$$

$$R_{mk}(\mathbf{p}) \geq R_{mk}^{\min}, \forall m, k, \tag{8d}$$

$$u_m f_m^{\mathsf{ue}} \leq f_m^{\max}, \forall m, \tag{8e}$$

$$\sum_{m \in \mathcal{M}} v_{mk} f_{mk}^{\mathsf{mec}} \leq f_k^{\max}, \forall k, \tag{8f}$$

$$(1), (6), (7), \tag{8g}$$

where $\mathbf{u} \triangleq \{u_m\}, \forall m \in \mathcal{M}$; $\mathbf{v} \triangleq \{v_{mk}\}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}$; $\mathbf{p} \triangleq \{p_m\} \forall m \in \mathcal{M}$; $\mathbf{f} \triangleq \{f_m^{\mathsf{ue}}, f_{mk}^{\mathsf{mec}}\}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}$ in their feasible domains. In (8), constraints (8b) and (8c) indicate the value range of transmission power and offloading portions of the tasks. Constraint (8d) shows the QoS requirements in terms of the transmission rate. The computation capacity of UEs and MECs is presented in (8e) and (8f), respectively. Finally, constraints (1), (6), (7) are already mentioned in above subsections.

3 Proposed Solution

It is challenging to solve the problem (8) directly with conventional approaches, due to the non-smooth and non-convex objective function, (8a) as well as non-convex constraints, i.e., (8d), (6), (7). Therefore, we propose an alternative optimisation (AO) solution to deal with this problem. We first solve for the transmission power ($\mathbf{p}$) of UEs with given values of $\mathbf{u}, \mathbf{v}, \mathbf{f}$. Next, offloading portions are optimised with given $\mathbf{p}, \mathbf{f}$. Finally, the processing rate of UEs and MECs is solved to complete the AO-based algorithm. The following subsections provide the development of our proposed solution.

3.1 Optimal Transmission Power with Given $(\mathbf{u}, \mathbf{v}, \mathbf{f})$

For any given $(\mathbf{u}, \mathbf{v}, \mathbf{f})$, problem (8) can be expressed as

$$\underset{\mathbf{p}}{\text{minimise}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{tot}(\mathbf{p}) \right\} \tag{9a}$$

$$\text{s.t. } (8b), (8d), (6), (7). \tag{9b}$$

To deal with the problem (9), we first address the transmission rate by using the logarithmic inequality given in [10,11], which follows from the convexity of the function $f(x, y) = \log_2 \Big(1 + 1/xy\Big)$ as follows

$$f(x, y) = \log_2(1 + \frac{1}{xy}) \geq \hat{f}(x, y). \tag{10}$$

Given $\forall x > 0, \bar{x} > 0, y > 0, \bar{y} > 0$, by applying first-order Taylor's expansion for $f(x, y)$, we have $\hat{f}(x, y) = \log_2 \Big(1 + \frac{1}{\bar{x}\bar{y}}\Big) + \frac{2}{(\bar{x}\bar{y}+1)} - \frac{x}{\bar{x}(\bar{x}\bar{y}+1)} - \frac{y}{\bar{y}(\bar{x}\bar{y}+1)}$. Let (i) denote the i-th iteration and substituting $x = \frac{1}{p_m |\varphi_{m,k,m}|^2}$, $y = \mathcal{I}_m(\mathbf{p}) + \sigma_k^2$, $\bar{x} = x^{(i)} = \frac{1}{p_m^{(i)} |\varphi_{m,k,m}|^2}$, and $\bar{y} = y^{(i)} = \mathcal{I}_m(\mathbf{p}^{(i)}) + \sigma_k^2$, we can obtain the approximation of wireless transmission rate for task offloading from the m-th IoT to the k-th MEC in (2) as

$$R_{mk}(\mathbf{p}) \geq \hat{R}_{mk}^{(i)}(\mathbf{p}), \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \tag{11}$$

where

$$\hat{R}_{mk}^{(i)}(\mathbf{p}) = B\Big(\log_2 \Big(1 + \frac{1}{\bar{x}\bar{y}}\Big) + \frac{2}{(\bar{x}\bar{y}+1)} - \frac{x}{\bar{x}(\bar{x}\bar{y}+1)} - \frac{y}{\bar{y}(\bar{x}\bar{y}+1)}\Big). \tag{12}$$

As a result, the constraint (8d) is now equivalent to

$$\hat{R}_{mk}^{(i)}(\mathbf{p}) \geq R_{mk}^{\min}, \ \forall m \in \mathcal{M}, \forall k \in \mathcal{K}. \tag{13}$$

This is now a convex constraint.

Next, to deal with the latency constraint, we introduce new variables $\mathbf{r} \triangleq \{r_{mk}\}_{\forall m,k}$ satisfying $1/\hat{R}_{mk}^{(i)}(\mathbf{p}) \leq r_{mk}, \forall m, k$. Then, the objective function $T_m^{\mathsf{tot}}(\mathbf{p})$ can be upper-bounded as

$$T_m^{tot}(\mathbf{p}) \leq \hat{T}_m^{\mathsf{tot}}(\mathbf{r}) = \frac{v_m C_m}{f_m + \tilde{f}_m} + \max_{k \in \mathcal{K}} \left(\frac{u_{mk} C_m}{f_{mk}^{\mathsf{mec}} + \tilde{f}_k} \right) + \max_{k \in \mathcal{K}} \{ r_{mk} u_{mk} D_m \} \tag{14}$$

Consequently, we can express (6) and (7) as

$$\begin{cases} \hat{T}_m^{\text{tot}}(\mathbf{r}) \leq T_m^{max}, \forall m, & (15a) \\ v_m \frac{\theta}{2} C_m f_m^2 + \sum_{k \in \mathcal{K}} p_m r_{mk} u_{mk} D_m \leq E_m^{max}, \forall m, & (15b) \\ \frac{1}{\hat{R}_{mk}^{ul(i)}(\mathbf{p})} \leq r_{mk}, \forall m, k, & (15c) \end{cases}$$

As we can see the constraint (15b) is still non-convex; therefore, we apply following inequality

$$xy \leq \frac{1}{2}\left(\frac{\bar{y}}{\bar{x}}x^2 + \frac{\bar{x}}{\bar{y}}y^2\right), \tag{16}$$

with $x = p_m$, $\bar{x} = p_m^{(i)}$, $y = r_{mk}$, $\bar{y} = r_{mk}^{(i)}$, to iteratively express (15b) as

$$u_m \frac{\theta}{2} C_m f_m^2 + \sum_{k \in \mathcal{K}} \frac{1}{2}\left(\frac{r_{mk}^{(i)}}{p_m^{(i)}} p_m^2 + \frac{p_m^{(i)}}{r_{mk}^{(i)}} r_{mk}^2\right) v_{mk} D_m \leq E_m^{max}, \forall m, k, \tag{17}$$

which is now a convex constraint. Problem (9) is equivalent to the following convex problem:

$$\min_{\mathbf{p},\mathbf{r}} \max_{\forall m \in \mathcal{M}} \left\{\hat{T}_m^{\text{tot}}(\mathbf{r})\right\}, \tag{18a}$$

$$\text{s.t. } (8b), (15a), (13), (15c), (17). \tag{18b}$$

To solve problem (18), a CVX package in MATLAB is used [5]. The proposed power allocation procedure for solving problem (18) is summarised in Algorithm 1.

Algorithm 1. Proposed algorithm for solving power allocation in problem (18)
.

Initialisation: Set $i = 0$ and initial feasible point $\mathbf{p}^{(0)}$; set the tolerance $\varepsilon = 10^{-3}$ and the maximum number of iteration $I_{max} = 20$.
  repeat
    Solve problem (18) for the next feasible solution $(\mathbf{p}^{(i+1)})$;
    Update $i = i + 1$;
  until Convergence or $i > I_{max}$;
  Output: Optimal power allocation coefficients $(\mathbf{p}^*)$.

3.2 Optimal Offloading Portions with Given $(\mathbf{p}, \mathbf{f})$

In this subsection, we solve for the optimal offloading portion with fixed $\mathbf{p}, \mathbf{f}$. Here problem (8) solving for $(\mathbf{u}, \mathbf{v})$ can be expressed as

$$\min_{\mathbf{u},\mathbf{v}} \max_{\forall m \in \mathcal{M}} \left\{T_m^{\text{tot}}(u_m, v_{mk})\right\}, \tag{19a}$$

$$\text{s.t. } (1), (6), (7), (8c), (8e), (8f). \tag{19b}$$

It is clearly seen that problem (19) is a standard convex program with all linear constraints that can be solved efficiently by CVX.

3.3 Optimal Frequency of UEs and MEC Servers with Given $(\mathbf{u}, \mathbf{v}, \mathbf{p})$

In this last sub-problem, we solve for the optimal frequency values of UEs and MECs, $\mathbf{f}$ with given $(\mathbf{u}, \mathbf{v}, \mathbf{p})$. To do that, we can rewrite problem (8) as follows

$$\min_{\mathbf{f}} \max_{\forall m \in \mathcal{M}} \left\{ T_m^{\mathsf{tot}} \left(f_m^{\mathsf{ue}}, f_{mk}^{\mathsf{mec}} \right) \right\} \tag{20a}$$

$$\text{s.t. } (6), (7), (8e), (8f) \tag{20b}$$

As we can see from (20), the problem is also a convex program with respect to variables $\mathbf{f}$. Therefore, it can be solved efficiently by CVX to obtain optimal solutions for the frequency of UEs and MECs.

Based on the above development, we propose using an iterative optimisation algorithm to solve the computing resource optimisation of UEs and MEC servers. The iterative algorithm is provided as following Algorithm 2.

Algorithm 2. Proposed iterative optimisation algorithm for solving (8).

Initialisation: Set $i = 0$, generate initial points ($\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{p}^{(0)}$, $\mathbf{f}^{(0)}$); set the error tolerance $\varepsilon = 10^{-3}$, and the maximum number of iteration $I_{max} = 20$.
repeat
With given ($\mathbf{u}^{(i)}$, $\boldsymbol{\beta}^{(i)}$, $\mathbf{f}^{(i)}$), solve problem (18) for optimal power control coefficients ($\mathbf{p}^{(i+1)}$);
With given ($\mathbf{p}^{(i+1)}$, $\mathbf{f}^{(i)}$), solve problem (19) for optimal offloading policies ($\mathbf{u}^{(i+1)}, \mathbf{v}^{(i+1)}$);
With given ($\mathbf{u}^{(i+1)}$, $\mathbf{v}^{(i+1)}$, $\mathbf{p}^{(i+1)}$), solve problem (20) for optimal frequency ($\mathbf{f}^{(i+1)}$);
Update $i = i + 1$;
until Convergence or $i > I_{max}$;
Output: Optimal solutions of $\mathbf{u}^*$, $\mathbf{v}^*$, $\mathbf{p}^*$, $\mathbf{f}^*$.

4 Numerical Results

For simulations, the parameters are set as presented in Table 1. Simulations are run on MATLAB and the convex optimisation problems are solved by CVX package.

It is clear from the results in Fig. 2 that as the computation resource limitations at the UEs increase, the worst-case latency decreases. This is because when there are more local computation resources, there is a greater ability to process tasks locally rather than offloading them to MEC servers. As a result, there is

Table 1. Parameter settings of simulations [1,19].

Parameter	Value
Number antennas of AP	4
Channel pathloss	$140.7 + 36.7 \log_{10} d$
System bandwidth	10 MHz
Maximum transmission power	30 dBm
Noise power density	−174 dBm/Hz
Task data size	100 kB
Task complexity	$[600, 1200]$ cycles/bit
Maximum UE frequency	1 GHz
Maximum MEC frequency	20 GHz
Maximum latency requirement	1 s
UE energy budget	1.5 J
The effective capacitance coefficient	10^{-24} watt.s^3/cycle^3

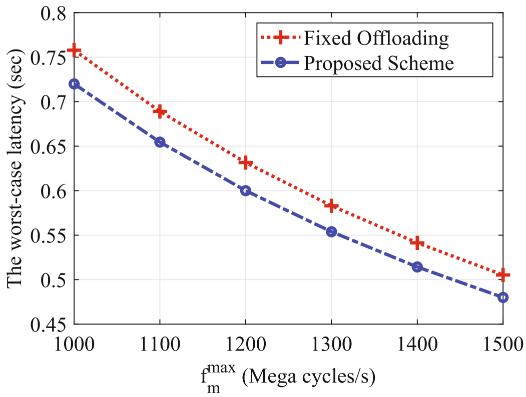

Fig. 2. Latency versus f_m^{max}.

a higher chance of reducing latency, especially when network resources are optimally allocated. The 'Proposed Scheme' uses the proposed solution written in this paper.'Fixed Offloading', uses fixed values for **u** and **v** to determine task offloading for all the UEs. The fixed values in this case are initialised values. It is clear that the optimised scheme must offload at a similar level to the initialised values, hence the difference in results of about 0.04 s.

Figure 3 demonstrates how the worst case latency changes with different values of UE task complexity (given in cycles per second). It is clear that as the task complexity increases, so does the worst case latency. This is due to both the UEs and MEC servers having to process a more complex task, which of course requires more time to do. Additionally the more complex tasks take more

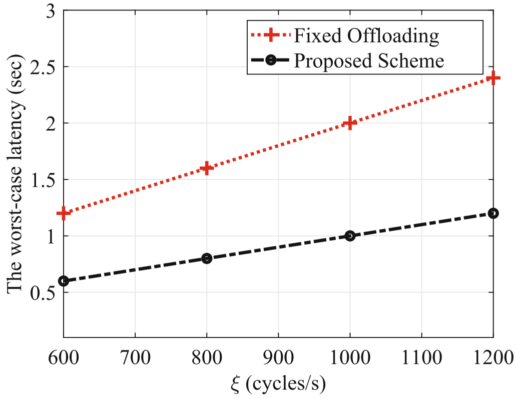

Fig. 3. Latency versus ξ.

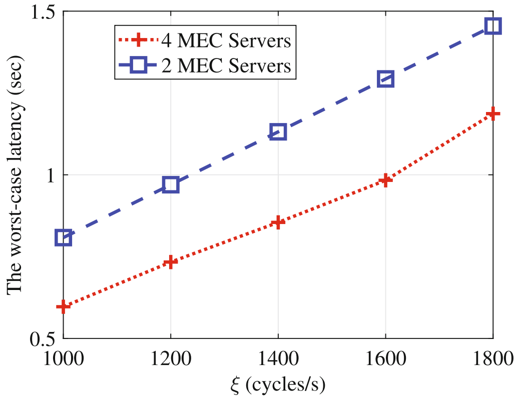

Fig. 4. Latency versus ξ with varying number of MEC Servers.

time to be offloaded, thus increasing the latency. What this figure demonstrates, is that there exists a predictable relationship, which could make guaranteeing performance simpler for end users. Once again, the proposed scheme performs better than a fixed scheme.

Figure 4 demonstrates the relationship between latency and the number of MEC servers. The worst case latency is lower with more MEC servers. As is made clear by both graphs, the latency increases as the task complexity increases, this graph also complements Fig. 3. The system with 4 MEC servers performs better to the system with 2 MEC servers.

5 Conclusion

In conclusion, we have presented a framework that assists in task offloading for IoT networks with mobile edge computing (MEC). Specifically, we have addressed the highly non-convex optimisation problem of minimising e2e latency in the system. The formulated problem has taken into account several variables such as transmission power, task offloading portions, CPU frequency of UEs and MECs subject to system budgets and QoS requirements. To evaluate the effectiveness of our proposed approach, we have conducted simulations by varying task complexity. Our simulation results clearly demonstrate that the proposed scheme outperforms the benchmark scheme, highlighting the efficacy of our approach in minimising e2e latency and improving the performance of IoT networks with MEC.

References

1. Do-Duy, T., Van Huynh, D., Dobre, O.A., Canberk, B., Duong, T.Q.: Digital twin-aided intelligent offloading with edge selection in mobile edge computing. IEEE Wirel. Commun. Lett. **11**(4), 806–810 (2022)
2. Duong, T.Q., Van Huynh, D., Li, Y., Garcia-Palacios, E., Sun, K.: Digital twin-enabled 6G aerial edge computing with ultra-reliable and low-latency communications: (invited paper). In: Proceedings of 1st International Conference on 6G Networking (6GNet), Paris, France (2022)
3. Elbamby, M.S., et al.: Wireless edge computing with latency and reliability guarantees. Proc. IEEE **107**(8), 1717–1737 (2019)
4. Elkashlan, M., Duong, T.Q., Chen, H.H.: Millimeter-wave communications for 5G - part 2: applications [guest editorial]. IEEE Commun. Mag. **53**(1), 166–167 (2015)
5. Grant, M., Boyd, S.: CVX: MATLAB software for disciplined convex programming, version 2.1, March 2014. http://cvxr.com/cvx
6. Ha, D.B., Truong, V.T., Lee, Y.: Performance analysis for RF energy harvesting mobile edge computing networks with SIMO/MISO-NOMA schemes. EAI Endorsed Trans. Ind. Netw. Intell. Syst. **8**(27), e2 (2021)
7. Li, Y., Huynh, D.V., Do-Duy, T., Garcia-Palacios, E., Duong, T.Q.: Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: a digital twin approach. IET Signal Proc. **16**(8), 897–908 (2022)
8. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutorials **19**(4), 2322–2358 (2017)
9. Merluzzi, M., Lorenzo, P.D., Barbarossa, S., Frascolla, V.: Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications. IEEE Trans. Signal Inf. Process. Netw. **6**, 342–356 (2020)
10. Nguyen, L.D., Tuan, H.D., Duong, T.Q., Dobre, O.A., Poor, H.V.: Downlink beamforming for energy-efficient heterogeneous networks with massive MIMO and small cells. IEEE Trans. Wirel. Commun. **17**(5), 3386–3400 (2018)
11. Nguyen, L.D., Tuan, H.D., Duong, T.Q., Poor, H.V.: Multi-user regularized zero-forcing beamforming. IEEE Trans. Signal Process. **67**(11), 2839–2853 (2019)
12. Nguyen, L.D., Kortun, A., Duong, T.Q.: An introduction of real-time embedded optimisation programming for UAV systems under disaster communication. EAI Endorsed Trans. Ind. Netw. Intell. Syst. **5**(17), e5 (2018)

13. Pan, Y., Chen, M., Yang, Z., Huang, N., Shikh-Bahaei, M.: Energy-efficient NOMA-based mobile edge computing offloading. IEEE Commun. Lett. **23**(2), 310–313 (2018)
14. Peng, K., Zhao, B., Qian, X., Xu, X., Zheng, L., Leung, V.C.M.: A multi-objective computation offloading method for hybrid workflow applications in mobile edge computing. In: Zhang, X., Liu, G., Qiu, M., Xiang, W., Huang, T. (eds.) CloudComp/SmartGift -2019. LNICST, vol. 322, pp. 47–62. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48513-9_4
15. Sun, W., Wang, P., Xu, N., Wang, G., Zhang, Y.: Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles. IEEE Internet Things J. **9**(8), 5839–5852 (2021)
16. Van Huynh, D., Khosravirad, S.R., Masaracchia, A., Dobre, O.A., Duong, T.Q.: Edge intelligence-based ultra-reliable and low-latency communications for digital twin-enabled metaverse. IEEE Wireless Commun. Lett. **11**(8), 1733–1737 (2022)
17. Van Huynh, D., Nguyen, V.D., Chatzinotas, S., Khosravirad, S.R., Poor, H.V., Duong, T.Q.: Joint communication and computation offloading for ultra-reliable and low-latency with multi-tier computing. IEEE J. Sel. Areas Commun. **41**(2), 521–537 (2023)
18. Van Huynh, D., Nguyen, V.D., Khosravirad, S.R., Duong, T.Q.: Fairness-aware latency minimisation in digital twin-aided edge computing with ultra-reliable and low-latency communications: a distributed optimisation approach (invited paper). In: 2022 56th Asilomar Conference on Signals. Systems, and Computers, pp. 1045–1049, Pacific Grove, CA (2022)
19. Van Huynh, D., et al.: URLLC edge networks with joint optimal user association, task offloading and resource allocation: a digital twin approach. IEEE Trans. Commun. **70**(11), 7669–7682 (2022)
20. Xiao, Y., Krunz, M.: Distributed optimization for energy-efficient fog computing in the tactile Internet. IEEE J. Sel. Areas Commun. **36**(11), 2390–2400 (2018)