Computer Communications 154 (2020) 361–369

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Joint offloading decision and resource allocation for mobile edge computing enabled networks

Yangzhe Liao, Liqing Shou, Quan Yu *, Qingsong Ai, Quan Liu

School of Information Engineering, Wuhan University of Technology, Wuhan, 430070, China

ARTICLE INFO

Keywords:
MEC
QoS
Resource allocation
Task offloading

ABSTRACT

Mobile edge computing (MEC) based solutions are essential and of great significance for a wide range of promising 5G wireless big data services such as remote healthcare systems and AR/VR games. Present research in this area focuses on the downlink resource allocation scenarios from MEC servers to user equipments (UEs). This paper considers a multi-user MEC-enabled wireless communication system, where UEs suffer limited communication and computation resources. To achieve higher energy efficiency and the better experience for UEs, we aim to maximize the number of offloaded tasks for all UEs in uplink communication while maintaining the computation resources of MEC at an acceptable level. The formulated problem is an NP-hard mixed-integer nonlinear programming problem and it is a challenge to solve it efficiently. As such, an efficient low-complexity heuristic algorithm is proposed, which provides a near-optimal solution with a low time cost. The results show that the proposed scheme achieves the higher number of successful offloaded tasks than the existing centralized resource allocation algorithm (CRAA) and centralized decision and resource allocation algorithm that UEs with the largest saved energy consumption accepted first (CAR-E) under different scenarios. Moreover, the relationship between the optimal transmission power and the computation resource of MEC is investigated. The results obtained in this paper can be extended to design a novel framework of communication, computation and smart coded caching MEC networks.

1. Introduction

The growing usage of smart user equipments (UEs), such as smartphones and wearable devices, has been developed rapidly in recent years, increasing resource-hungry services like 4K/8K high definition video, AR/VR online games and remote e-health monitoring systems have come into life [1–3]. These novel applications have the trend of attracting significant attention from the users and bring high profits of the mobile operators. However, this may bring a high load on both the UEs and the communication network. Moreover, owing to the constraint of physical size, UEs are struggling to handle delay-sensitive and computation-intensive applications due to lack of communication or computation resources.

Mobile cloud computing (MCC) was proposed to provision UEs by offloading computation-intensive tasks to remote commercially Cloud platforms such as Amazon EC2 and ThinkAir via uplink and receive the results via downlink [4,5]. By doing so, the majority of computation loads are transformed from battery-charged UEs to the Cloud, and thus can significantly reduce the energy consumption of UEs and prolong the network lifetime. However, there exist numerous technical difficulties for MCC to support UEs. The authors in [6] formulated the offloading decision problem for MCC among all UEs as a decentralized computation offloading game. Although the Cloud-based computation method can expand the computation capability of UEs to some degree, developing a reliable offloading strategy for delay-sensitive services remains challenging. One of the key challenges is that public Cloud is generally placed a few thousand kilometers away from UEs, offloaded task transmission suffers high transmission delay [7]. Additionally, if too many UEs choose to offload the tasks to the Cloud simultaneously, it is highly like to generate severe interference and thus decrease the overall network performance [8].

Fortunately, the emergence of mobile edge computing (MEC) communications aims to make mobile operation greener and more efficient [9–11]. MEC is placed within the radio access network (RAN), as an expansion in the evolution of 5G communications and offers computation resources at the edge of the network. The advantages of this emerging technology consist of low latency, reliable service delivery, highly efficient network operation and so forth. All of the above advantages bring new opportunities for mobile operators to perform profitable services, allowing them to deliver latency-sensitive services to UEs and receive computation-intensive workload from UEs [12]. By integrated with remote radio heads (RRHs) at the numerous types of existing base stations (BSs) such as Pico and Femto, delay-sensitive

* Corresponding author.
E-mail address: yuquan@whut.edu.cn (Q. Yu).

https://doi.org/10.1016/j.comcom.2020.02.071
Received 15 December 2019; Received in revised form 17 January 2020; Accepted 23 February 2020
Available online 29 February 2020
0140-3664/

and computation intensive tasks can be offloaded to edge Cloud rather than remote public Cloud, and thus decrease the transmission delay and improve UEs' experience. Moreover, base-band units (BBUs) in MEC is in charge of receiving the offloaded tasks from UEs via the uplink and sending executed results back to UEs via the downlink [13]. After receiving the offloading requests, MEC allocates the virtual machines (VMs) located in BBU pool that provide computation resources to serve UEs. Furthermore, BBUs are connected with RRHs via optical fiber that is capable of exchanging information between the edge Cloud and RRHs, which achieves low transmission delay. Other advantages of MEC including proximity, location awareness and so forth. Consequently, MEC-enabled communications bring Cloud-like wireless service to UEs and also rise the profit for mobile operators. As reported in [14], MEC is just not only a service provider for UEs but also has the potential to become the next generation computing service provider. To realize the promising technique, some existing technical challenges are listed as follows along with selected up to date references.

1.1. Task scheduling

Some efforts have been made on task scheduling and resource allocation for MEC [7,15–18]. In [7], the proposed a MEC-enabled system model considers only one UE sending the offloading request. The authors in [15] proposed a novel offloading decision algorithm to analyze a number of UEs sending the offloading requests simultaneously. However, transmission interference among UEs is ignored. The authors in [16] analyzed independence among different types of UEs in their offloading decisions under infinite computational capabilities on MEC with uplink interference. This is because MEC not only holds the information of computing resources but also has the wireless channel status of all UEs within the coverage. The authors in [17] considered the average time cost of all generated tasks and formulated the task scheduling problem as an energy-constrained delay minimization problem. The problem then is proved that can be solved by employing an efficient one-dimensional search algorithm rather than the existing two-timescale stochastic optimization method. [18] demonstrated that it is challenging to obtain an optimal offloading policy for a dynamic network. The authors transformed the original problem as the sum cost of delay and energy consumption of all UEs and then proposed a deep reinforcement learning-based method.

1.2. Resource allocation

Some work focus on joint optimization of task offloading decisions and resource allocation of all UEs, typically for delay-tolerant services have been studied in [19–23]. In [19], the authors proposed a jointly computing resource and transmission bandwidth strategy. Some UEs that suffer high interference may increase the transmission power to promise the link quality. However, this may increase the interference among UEs and lead to the offloading failure of other UEs. It is worth noticing that graph theory is a useful tool to solve resource allocation in MEC [24]. The authors in [25] employed the graph to represent the current network topology. The results obtained by graph theory then can instruct the MEC server to find a candidate link for offloading that can achieve satisfactory performance. However, the graph-based methods cannot sufficient illustrates the relationship between UEs and channels. As demonstrated in [26], the hypergraph based scheme outperforms the traditional graph when investigating the cumulative interference in wireless communications. In this aspect, hypergraph theory is proposed as an effective tool to investigate cumulative interference and to model the complicated relations among channels. However, due to the nature of the interference, it is a challenge to define the weight for each hyperedge [26,27]. An iterative hypergraph matching algorithm is proposed in [28]. In each iteration, each hyperedge will update its weight and the selected hyperedges cannot have a common channel vertex, i.e., a channel only can be allocated with one UE. We find that hypergraph may be applicable to solve resource allocation problems for MEC regarding effectively guide UEs to make the offloading decision and channel allocation.

Another research challenge is that mobile operators suffer a limited amount of wireless communication resources such as bandwidth and computation resources. The network performance may get worse when a large number of UEs are sending offloading tasks simultaneously. Moreover, this may also occur severe interference and lead to inefficient use of wireless resources if there is no suitable central coordinating control mechanism. Besides, handling computation-intensive or high bandwidth required tasks may increase the operating expense of the mobile operators, severely decreases the profit and also still cannot meet the strict QoS requirements of UEs. Therefore, task offloading benefit is of considerable significance when design the access control mechanisms. How to maximize the number of offloaded tasks while maintaining the overall cost of MEC at an acceptable level still needs further investigation.

To realize the higher energy efficiency and the better experience for UEs, computation offloading strategies for MEC have been widely investigated in the literature recently. On the one hand, due to the dynamic of task generation, UEs may dynamically change the offloading strategy. On the other hand, since computation resources in MEC are limited by the number of available VMs, admission control is of great significance in managing the offloaded tasks because some tasks may not achieve offloading benefits and occupy bandwidth resources [14]. Moreover, some advanced techniques, i.e., the dynamic voltage and frequency (DVFS), are capable of adjusting CPU-cycle frequency to reduce the system cost regarding execution time and energy consumption [11]. With the DVFS technique, the energy consumption of local execution can be minimized by dynamic adjust the CPU-cycle. Besides, some work such as [17] analyzed the cost tradeoff between energy consumption and execution time by employing Lyapunov optimization, which combines CPU frequency and resource allocation as a two-stage problem.

To the best of our knowledge, joint decision making and communication computation resource allocation for multiuser offloading networks considering interference have not yet fully investigated. The majority of current research is focused on downlink transmission research while very few work focus on uplink offloading benefit analysis. When MEC's communication and computation resources are limited, task admission control is proved as a non-deterministic polynomial hard (NP-hard) mixed-integer non-linear programming problem in literature [6] and is quite challenging to solve it. Furthermore, since the offloading tasks or offloading decisions may be updated after receiving the feedbacks from MEC, further efforts should be made regarding bandwidth resource reallocation.

1.3. Main contributions

In this paper, we aim to maximize the number of successfully offloaded tasks for battery-charged UEs to reduce power consumption while maintaining network performance at a satisfactory level. To solve the formulated NP-hard mixed-integer nonlinear programming problem, a low-complexity heuristic algorithm is proposed. We first consider the interference among all UEs and employ hypergraph approaches to investigate channel pre-allocation. Then, task offloading is analyzed jointly with MEC computation and communication resource allocation. Assume the uplink and downlink duality, the optimal transmission power is obtained. After receiving the feedback from MEC, UEs will adjust the offloading decisions and bandwidth will be reallocated to improve the network performance. The relationship between the number of successfully offloaded tasks and the number of UEs under different computation resources of MEC are investigated in detail. The results show that our proposed scheme achieves a higher number of offloaded tasks than the selected existing techniques, i.e., centralized resource allocation algorithm (CRAA) in [29] and centralized decision

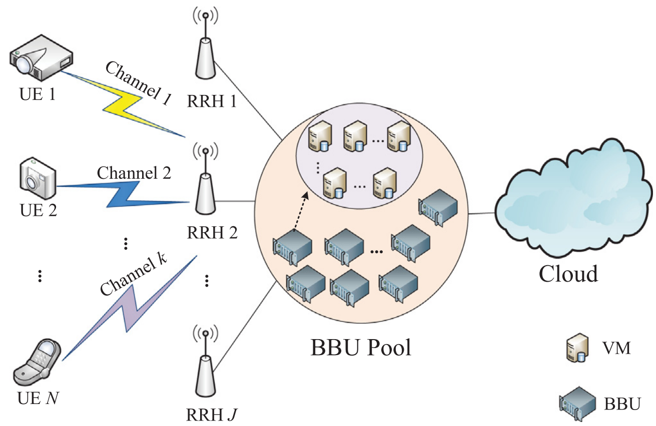

Fig. 1. The proposed MEC-enabled network.

and resource allocation algorithm that UEs with largest saved energy consumption accepted first (CAR-E) [30]. Moreover, the relationship between the optimal transmission power and the computation resources of MEC is given.

The rest of this paper is organized as follows: Section 2 presents the system model and problem formulation. The proposed solutions to the problem are given in Section 3. Section 4 demonstrates the performance evaluation and Section 5 concludes the paper and identifies some potential research points in this field.

2. System model

With the highly increasing demand for wireless big data services, MEC-enabled wireless communications have attracted significant attention. As shown in Fig. 1, consider there are $\mathcal{N} = \{1, 2, \dots, N\}$ UEs that are capable of offloading tasks to MEC, either computation-intensive or delay-sensitive, and k channels are available to use. To this regard, the traditional BS is divided into three significant components, i.e., RRHs, the uplink and downlink, and the BBU pool. BBUs are generally placed in the edge data center to organize a BBU pool, which can be efficiently managed by centralized processing. Moreover, this novel system architecture allows distributed low-power RRHs connecting the BBU pool to perform as relays to receive the offloaded tasks from UEs and forward the result to UEs from BBU pool. After UE i sending its offloading request, MEC assigns an RRH j, $j \in \mathcal{J} = \{1, 2, \dots, J\}$ to serve the UE, which forwards the computation task to a corresponding VM. Assume that each RRH has a corresponding VM, the transmission time between RRHs and MEC servers, e.g., VMs, is ignored in this paper since RRHs and BBU pool are connected via optic fiber. The offloaded task from UE i can be executed by VM j located in the BBU pool. One should note that MEC has the channel state information (CSI) while UEs do not have any information regarding other UEs.

Definition 1. Let UE i has the computational task U_i and can be denoted by $U_i = \left(F_i, D_i, T_{i,\max}\right), i \in \mathcal{N}$, where F_i and D_i denote the total number of CPU cycles and offloaded data size, respectively, and $T_{i,\max}$ is the maximum time allowance to execute the task. All generated tasks can be divided into three different sets, namely the offloading set $\mathcal{O}$, the local execution set $\mathcal{L}$ and the reschedule set $\mathcal{R}$.

Consider each UE has a computation task that needs to be executed. The offloading decision variable, $\alpha_{ij}, i \in \mathcal{O}, j \in \mathcal{J}$ to denote whether UE i decides to offload a task U_i to jth place for further execution, e.g., RRH j (the corresponding VM j in MEC, $j \neq 0$), one can obtain that

$$C1 : \alpha_{ij} \in \{0, 1\}, i \in \mathcal{O}, j \in \mathcal{J}, \tag{1}$$

where $\alpha_{ij} = 1$ means that UE i decides to offload its task to VM j and $\alpha_{ij} = 0$ indicates UE i decides not to offload the task U_i. Moreover, we assume that all tasks can only be executed in one place. It is possible that a task cannot be completed anywhere within the maximum time allowance $T_{i,\max}$ due to lack of communication or computation resources, and thus one has

$$C2 : \sum_{j \in J} \alpha_{ij} = 1, i \in \mathcal{O}. \tag{2}$$

2.1. Communication and computation models

If UE i decides to offload U_i to VM j, let p_i^{tr} is the transmission power of UE i and should follow as

$$C3 : p_i^{tr} \le p_{i,\max}, \tag{3}$$

where $p_{i,\max}$ is the given maximum transmission power of UE i. Similarly, the uplink bandwidth allocation should follow as

$$C4 : \sum_{i \in \mathcal{N}} B_{ij} \le B_{ij,\max}, \tag{4}$$

where B_{ij} is the bandwidth between the UE i and RRH j. $B_{ij,\max}$ is the maximum bandwidth that can be allocated to UEs. Moreover, the time cost to offload the task can be expressed as

$$T_{ij}^{o} = \frac{D_i}{r_{ij}}, \tag{5}$$

where r_{ij} is the transmission data rate from UE i to its corresponding VM j. The time of VM j to execute the task can be given as

$$T_{ij} = \frac{F_i}{f_j}, \tag{6}$$

where f_j is the computation capacity of VM j. If UE i decides to offload the task, one can obtain the time cost constraint as

$$C5 : \sum_{j \in \mathcal{J}} \alpha_{ij} \left(\frac{D_i}{r_{ij}} + \frac{F_i}{f_j} \right) \le T_{i,\max}. \tag{7}$$

The computing power consumption of VM j can be expressed as

$$p_j = \kappa_j \left(f_j\right)^{\zeta_j}, \tag{8}$$

where κ_j and ζ_j represent the switched capacitance and positive constants, respectively, which are dependent on hardware. For UE i that decides to offload a task, the transmitted signal can be expressed as

$$s_i = \sqrt{p_i^{tr}} b_i, \tag{9}$$

where b_i is the transmitted data symbol. Assume that b_i is with unity average power and $\mathbb{E}\left(|b_i|^2\right) = 1$. MEC will decide to accept or reject the offloading requests according to the computation and communication resources. Assume MEC's service computing unit and communication computing unit are limited, they are normally constrained by the physical size of VMs. Let the maximum computation resource of VM j to service UEs as $F_{j,\max}^{C}$, one can see the computation constraint

$$C6 : \sum_{i \in O} f_j \le F_{j,\max}^{C}. \tag{10}$$

Similarly, if we assume the maximal computing resource allocation to support communication computing as $F_{j,\max}^{B}$, one can obtain the communication constraint of jth place as

$$C7 : \sum_{i \in O} r_{ij} \le F_{j,\max}^{B}. \tag{11}$$

2.2. Problem formulation

Owing to the technical constraints of UEs' batteries, the power supply is a major bottleneck for the 5G wireless big data services. In this paper, we aim to maximize the total number of offloaded tasks to prolong the network lifetime, e.g.,

$$\begin{aligned} \mathcal{P}1 : \quad & \max_{\alpha_{ij}, p_i^{tr}, B_{ij}} \quad |\mathcal{O}| \\ & \text{s.t. } C1\text{–}C7 \end{aligned} \tag{12}$$

where α_{ij} is binary variable, p_i^{tr} and B_{ij} are continuous variables.

Assumption 1. $\mathcal{P}1$ is a mixed-integer nonlinear programming problem. This type of problem is NP-hard and cannot be effectively solved by existing optimization techniques. We admit that this problem may be feasible by utilizing the exhaust search algorithm at a high complexity cost. In this paper, we propose an algorithm to decompose the original problem into a two-stage problem and then solve them separately using the iterative method until convergence. The original problem can be effectively solved in polynomial time by designing the following steps:

Step 1: Channel Pre-allocation. We assume that multiple UEs can share one channel to improve spectrum efficiency. Due to the physical size constraints of UEs, we first analyze the channel pre-allocation based on interference and local information, UEs that suffer high interference are encouraged to offload its task to MEC to reduce the overall network interference, e.g., move to offload set $\mathcal{O}$. For those UEs that cannot assign bandwidth will be moved to local execution set $\mathcal{L}$.
Step 2: Task Offloading Benefit Analysis. According to Step 1, only UE $i \in \mathcal{O}$ can achieve offloading benefit will allow sending offloading requests to MEC for further investigation. For tasks that cannot benefit from task offloading will be moved to the reschedule set $\mathcal{R}$. Task offloading benefit is proposed based on the selected parameters such as task execution delay and energy consumption.
Step 3: Selected Parameters Update. The key challenge of downlink resource allocation is to guarantee user fairness. After receiving the results from RRHs, all UEs will follow the instructions and update the offloading set $\mathcal{O}$ and bandwidth reallocation.

3. The proposed low-complexity heuristic algorithm

In this section, the solution to the proposed problem $\mathcal{P}1$ is given in detail in the following step: first, we utilize the graph theory to realize the bandwidth pre-allocation while decreasing the overall interference. Second, for UEs that are allocated channels are encouraged to offload tasks to MEC. However, due to the limitation of the transmission power of UEs and communication and computation resources of MEC, not all offloading requests can be accepted by MEC. We first propose the uplink channel pre-allocation method based on a hypergraph-based technique. The resource allocation is then analyzed by jointly consider numerous parameters, e.g., the task offloading benefit, the computation resource of MEC and the downlink channel allocation.

3.1. Bandwidth pre-allocation scheme

Before considering the task offloading strategy, we analyze the channel pre-allocation, the received signal at RRHs can be given as

$$\mathbf{y} = \sum_{i\in\mathcal{O}} \boldsymbol{h}_i \sqrt{p_i^{tr}} b_i + \mathbf{z}, \tag{13}$$

where $\boldsymbol{h}_i$ is the CSI from an offloading request UE i to all RRHs and $\mathbf{z}$ is the additive white Gaussian noise vector, which follows $\mathcal{CN}\left(0,\sigma^2\right)$. The instantaneous SINR of the received signal at MEC from UE i over uplink channel k can be given as

$$SINR_i^{up} = \frac{p_i^{tr} g_i}{\sigma^2 + \sum_{\mathcal{K}\backslash k, n\neq i} p_n^{tr} g_n}, i, n \in \mathcal{N}, i \neq n, \tag{14}$$

where $\sum_{\mathcal{K}\backslash k,n\neq i} p_n^{tr} g_n$ is the interference from other UEs belong to $\mathcal{O}\backslash\{i\}$. g_i and g_n represent the channel from UE i and UE n, $n \in \mathcal{O}\backslash\{i\}$ to MEC. σ^2 is variance of the noise. Interference management is one of the critical issues to analyze the channel resources since UEs share the same channel may lead to mutual interference. Unlike some existing work [27,31], we utilize a hypergraph-based solution to deal with pre-bandwidth allocation. We formulate the channel resources as k different colors. To construct the hypergraph, UEs are represented by vertices and interference are denoted by hyperedges. Color set C represents the set of channels, and has k different colors. To avoid mutual interference, it is not difficult to check that same hyperedge cannot share the same color. Moreover, to balance the network capacity and the computational complexity, the number of UEs in a hyperedge will be dynamically adjusted as the network conditions change. If UE i can form a hyperedge, i.e.,

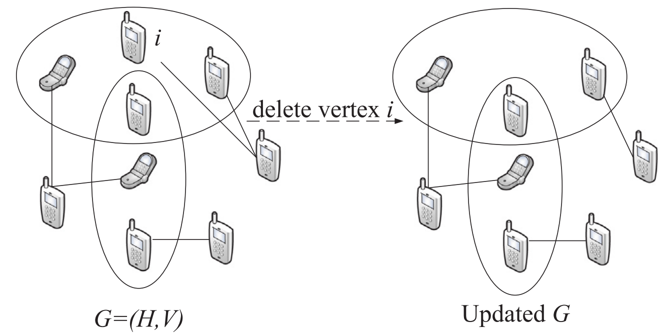

Fig. 2. An example of a deletion of hyperedges.

$$SINR_i^{up} \geq \gamma_i^k, \tag{15}$$

where γ_i^k is the predefined interference threshold over channel k. The interference is defined as the initial weight of a hypergraph. Find a vertex i of the maximum monodegree in the hypergraph, delete hyperedges that contains vertex i. If interference is reduced after the operation, this hyperedge is deleted and the weights of the hyperedges will be updated until color set C is empty.

In a hypergraph $G(H,V)$ where H and V represent the hyperedge set and the vertex set, respectively. A weak deletion of a vertex $v, v \in V$ is to delete v from V and update the hyperedge as $h_1 = h\backslash\{v\}, h \in H$. An example of weak deletion a vertex from a hypergraph given in the following example.

Example. Hypergraph coloring has been widely used in many-to-one assignment problems. The hypergraph coloring is a centralized based scheme that MEC has the global information of the hypergraph and selects a color for each vertex. As shown in Fig. 2, there exists numerous hyperedges in one hypergraph. A deletion hyperedge $v \in \mathcal{V}$ means to remove the hyperedge v from the original hypergraph G with other elements remained. One can use $G_1 = G\backslash v$ as the weak delete operation of G.

The monodegree $m(v,H)$ of vertex $v \in V$ in a hypergraph $G(H,V)$ can be regarded as the maximum cardinality of all hyperedges. For example, a hypergraph $G_1(v)$ has two elements, e.g., $e_i, e_n \in G_1(v)$ and $e_i \cap e_n = \{v\}$. Let G_Y is a sub-hypergraph of the hypergraph, the first vertex to color is the one with the highest value of $\max_{Y\in V} \min_{v\in Y} m(v, H_Y)$. Moreover, the search for the next coloring vertex is to find the maximum interference path. The aim is to achieve the minimal interference among all UEs until the available color set become empty. The bandwidth pre-allocation scheme can be divided into the following steps. First, the hypergraph can be constructed according to the SINR, edges can be formed when the uplink SINR is not less than the predetermined SINR γ_i^k, i.e., $SINR_i^{up} \geq \gamma_i^k$. Moreover, channel allocation is considered, the weights of all hyperedges can be updated based on the interference. Then, graph coloring will be implemented until the available color set C become empty. As a result, one can roughly obtain the offloading set $\mathcal{O}$, the pre-allocation bandwidth B_{ij} and the offloading decision α_{ij}. One should note that for those vertices that are still uncolored, there are no channel resources for them to send offloading requests to MEC at this stage. Therefore, some selected UEs are assigned with bandwidth pre-allocation and can send offloading requests to MEC for further investigation. The detailed information regarding the hypergraph-based pre-allocation scheme can be found in Algorithm 1.

Algorithm 1 The Bandwidth Pre-allocation Algorithm.

Inputs:
Vertices: UEs
Hyperedges: interference
Color set: available channels $\mathcal{C}$
Step 1: Construct the hypergraph
If $SINR_i^{up} \geq \gamma_i^k$
  Form edges
  Until all UEs checked
Else
  $\alpha_{ij} = 0$
  i move to $\mathcal{L}$ //move to local execution set
Hypergraph construction finished
Step 2: Channel allocation scheme
  Define the interference as the initial weight of a hyperedge.
Find a vertex i of the maximum monodegree
For $i = i - 1$
  delete those hyperedges that have the same UEs
  $H_{i-1} = H_i - x_i$
  If interference is reduced, this hyperedge is deleted.
   Update weights of hyperedges
  Else
   Continue
Until hypergraph has an empty hyperedge set
Step 3: Graph coloring
$x_i = \max_{Y \in V} \min_{v \in Y} m(v, H_Y)$
set $i = i + 1$
Find $i = \arg\max_i x_i$
$\alpha_{ij} = 1$
Until available color set $\mathcal{C}$ is empty
remain the vertex uncolored.
Outputs: Roughly decided $\mathcal{O}, B_{ij}, \alpha_{ij}$

Proposition 1. *Let Q is the number of UEs within the same hyperedge, which significantly influences the computation complexity of Algorithm 1. Moreover, the network capacity will increase with the higher value of Q while suffers higher computation complexity.*

Proof. For a hypergraph-based coloring scheme, denote the number of UEs within the same hypergraph is Q, the computation complexity of the interference is proportional to $O(Q^2)$, and it is necessary to check $(Q-1)*(Q-2)$ hyperedges. The computational complexity of channel allocation can be roughly given as $C_{\text{graph}} \propto O(Q^3)$. The value of Q is optional according to the network conditions. With a higher value of Q, the complexity $O(Q^3)$ to construct a hypergraph is becoming extremely high as proved in [32]. However, the network capacity depends on the number of channels, and will not significantly improve with the increase of Q. To achieve a compromise between network capacity and computational complexity, the number of Q is limited within 2, e.g., $Q \leq 2$. The hypergraph-based channel pre-allocation problem can be solved in polynomial time. ■

3.2. Resource allocation analysis

In this part, we start to analyze the offloading strategy for UEs. Due to the limitation of uplink bandwidth and computational resources of MEC, not all offloading requests can be accepted by MEC. Assume UE i that can benefit from task offloading, since different VMs have different computation capacities. For UEs that are interested in offloading tasks, one can expect that

$$\sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{O}} f_j \leq F^c, \tag{16}$$

where F^c is the maximum number of BBU pool that can serve UEs. The original problem can be transformed into the following subproblems, e.g., $\mathcal{P}1.1$ and $\mathcal{P}1.2$. Since not all offloading requests obtained from Algorithm 1 can be successfully executed by MEC, to maximize the total number of offloaded tasks, we first analyze $\mathcal{P}1.1$, which is to maximize the total number of successful offloaded tasks

$$\begin{aligned} \mathcal{P}1.1: \quad & \max_{p_i^{tr}} \ |\mathcal{O}| \\ \text{s.t.} \quad & C3, C6\text{–}C7 \\ & C8: \textstyle\sum_{j\in J} \alpha_{ij}\left(\frac{D_i}{r_{ij}} + \frac{F_i}{f_{ij}}\right) \leq T_{i,\max}, i \in \mathcal{O} \end{aligned} \tag{17}$$

where $\mathcal{O}$ is obtained from Algorithm 1. The optimization variable p_i^{tr} is constrained with $C3$ and $C8$. Moreover, $C6$ and $C7$ are constrained by MEC, which significantly influence the successful ratio of task offloading. We notice that this problem is still challenging to be solved effectively. These parameters still need further investigation, e.g., update $\mathcal{O}$ in the next following section.

Remark 1. The optimization problem $\mathcal{P}1.1$ is NP-hard and challenging to be solved. One should note that due to the constraint $C8$, the optimization problem $\mathcal{P}1.1$ is non-convex.

Since UE $i, i \in \mathcal{O}$ do not have global information regarding other UEs, we first consider UEs that can benefit from task offloading as mentioned in Proposition 2. The corresponding optimal power consumption can be expressed as $p_i^* = \kappa_i \left(\frac{F_i}{T_{i,\max}}\right)^{\xi_i}$ where κ_i is the effective switched capacitance and ξ_i is the positive constant. One should note that f_i^* and p_i^* are only feasible when $f_i^* \leq f_{i,\max}$. When $f_i^* > f_{i,\max}$, there is no solution. This is due to the maximum computing frequency of UE i that cannot execute within the time allowance by itself. For UE $i, i \in \mathcal{O}$ that cannot benefit from task offloading will be moved from $\mathcal{O}$ to $\mathcal{R}$. As a result, the variable α_{ij} is solved efficiently and tasks that can benefit from task offloading is updated from $\mathcal{O}$ to $\mathcal{O}^H$, which is the set of UEs that can benefit from task offloading. Moreover, recall Eqs. (5) and (6), the minimum transmission data rate can be given as

$$r_{ij,\min} = \frac{D_i}{T_{i,\max} - \frac{F_i}{f_j}}. \tag{18}$$

Proposition 2. *If single-antenna UE $i, i \in \mathcal{O}$ is equipped with single-antenna that can benefit from task offloading, the minimal power consumption should be*

$$p_i^{tr*} = \kappa_i \left(\frac{F_i}{T_{i,\max}}\right)^{\xi_i}, \tag{19}$$

where κ_i and ξ_i are only depend on UE's hardware.

Proof. The transmission symbols are with unity average for all UEs as $\mathbb{E}(s_i)^2 = 1, i \in \mathcal{O}$. By calculating the derivation of E_i^{tr}, one can obtain that $\left(\frac{E_i^{tr}}{p_i^{tr}}\right)' \geq 0$. Since the decrease of E_i^{tr} leads to p_i^{tr} decrease correspondingly. That is, for each UE that transmits the offloading request to RRH, minimizing UE's energy consumption is equivalent to minimize power consumption. One can see that due to the time cost constraint of the time is $T_{i,\max}$, we can obtain the optimal computational frequency as $f_i^* = \frac{F_i}{T_{i,\max}}$. ■

Consider UEs belong to $\mathcal{O}^H$, to minimize the power consumption of UE $i, i \in \mathcal{O}^H$, the uplink power consumption optimization problem can be formulated as

$$\begin{aligned} \mathcal{P}1.2: \quad & \min_{p_i^{tr}} \ \textstyle\sum_{i\in\mathcal{O}^H} p_i^{tr} \\ \text{s.t.} \quad & C3, C6\text{–}C7 \\ & C9: r_{ij} \geq r_{ij,min} \end{aligned} \tag{20}$$

When offloading requests are received by the MEC via the predetermined channels as obtained from Algorithm 1, the execution priority of tasks at VM j is determined according to the remaining time of tasks before they exceed the maximum time allowance, i.e., $T_{i,\max} - T_i^o$. Then we will check whether MEC resource constraints $C6$ and $C7$ can be satisfied, i.e., if the computation and communication resources provided by MEC can meet the QoS requirements of $\mathcal{O}^H$. One can transform the original problem $\mathcal{P}1.2$ into $\mathcal{P}1.3$ by relaxing the constraints $C6$ and $C7$.

$$\begin{aligned} \mathcal{P}1.3: \quad & \min_{p_i^{tr}} \quad \sum\nolimits_{i\in\mathcal{O}^H} p_i^{tr} \\ \text{s.t.} \quad & \\ & C3, C9 \end{aligned} \tag{21}$$

One can see that each UE can only be achieved lower transmission delay by increasing the transmission power p_i^{tr}. Assume $\mathcal{O}^H$ consists of N offloading requests, to meet the constraint $C9$, we formulate the minimal data rate as $\boldsymbol{r}_{min} = \left[r_{1j}^{min}, \ldots, r_{Nj}^{min}\right]$, which lay on the lower boundary of an N-dimensional achievable data rate region with the sum power constraint denoted by $\|\mathbf{p}^{\mathrm{tr}}\|_1$. Therefore, the optimal power p_i^{tr} is obtained by move tasks that cannot meet the constraint $C3$ from $\mathcal{O}$ to $\mathcal{R}$. Define a new set $\mathcal{A}$ that includes UEs exceed the maximum resource of the MEC when those tasks in $\mathcal{O}^H$ that cannot meet $C6$ constraint as

$$\begin{aligned} \mathcal{A} = \Bigg\{ & i > m \Big| \sum_{i=1}^{m} f_{ij} \le F_{j,\max}^{C} \& \\ & \sum_{i=1}^{m+1} f_{ij} > F_{j,\max}^{C}, i \in \mathcal{O}^H, j \in \mathcal{J} \Bigg\}. \end{aligned} \tag{22}$$

Meanwhile, one can transform constraint $C7$ as

$$\begin{aligned} \mathcal{B} = \Bigg\{ & i > n \Big| \sum_{i=1}^{n} r_{ij} \le F_{j,\max}^{B} \& \\ & \sum_{i=1}^{n+1} r_{ij} > F_{j,\max}^{B}, i \in \mathcal{O}^H, j \in \mathcal{J} \Bigg\}. \end{aligned} \tag{23}$$

One can obtain the $\mathcal{F} = \mathcal{A} \cup \mathcal{B}$. If $\mathcal{F} \neq \emptyset$, UE $i \in \mathcal{A}$ can be moved from $\mathcal{O}^H$ to $\mathcal{L}$ for local execution or reschedule in the next round until $\mathcal{F} = \emptyset$. By utilizing the uplink–downlink duality, the optimization problem of downlink power consumption can be given as

$$\begin{aligned} \mathcal{P}1.4: \quad & \min_{\boldsymbol{m}_j} \quad \sum\nolimits_{i\in\mathcal{O}'} \boldsymbol{m}_j^H \boldsymbol{m}_j \\ \text{s.t.} \quad & \\ & C10: r_j \ge r_{j,min} \\ & C11: \sum\nolimits_{i\in\mathcal{O}'} \left\| \left\| m_j \right\|^2 \right\|_1 \le F_{j,\max}^{C} \\ & C12: \sum\nolimits_{i\in\mathcal{O}'} r_j \le F_{j,\max}^{B} \end{aligned} \tag{24}$$

where $\mathcal{O}'$ is obtained by updating $\mathcal{O}^H$ and $\mathbf{m}_j$ is the downlink transmission beamforming vector from all RRHs to UE i. r_j is the transmission data rate from MEC to UE i. The downlink transmission rate can be given as

$$r_j = B_j \log_2 \left(1 + SINR_j^{down}\right), \tag{25}$$

where $SINR_j^{down} = \frac{p_j^{tr} \left\| \boldsymbol{m}_j^H \boldsymbol{h}_j \right\|^2}{\sigma^2 \left\| \boldsymbol{m}_j \right\|^2 + \sum_{\mathcal{K}\setminus k, j\neq i} p_i^{tr} \left\| \boldsymbol{m}_j^H \boldsymbol{h}_j \right\|^2}$, $\boldsymbol{m}_j$ and $\boldsymbol{h}_j$ represent the transmission beamforming vector and the CSI for RRH j to a corresponding UE, respectively. B_j is the bandwidth between RRHs and UE i. Since downlink resource allocation problem has been widely investigated, we utilize the same manner proposed in [29]. As such, the problem $\mathcal{P}1.3$ can be transformed to the second-order cone (SoC) constraint in the downlink resource allocation as proved in [33]. Moreover, by applying uplink and downlink duality, one can transform the constraint in $C10$ to the second-order cone (SoC), e.g.,

$$\begin{aligned} C13: & \sqrt{1 - \frac{1}{2^{\frac{D_i}{B_i T_i}}}} \sqrt{\sum\nolimits_{i\in\mathcal{O}'} \left| \sum\nolimits_{j\in\mathcal{J}} h_{ij}^H v_{kj} \right|^2 + \sigma^2} \\ & \le \mathrm{Re}\left(\left| \left| \sum\nolimits_{i\in\mathcal{O}'} h_{ij}^H v_{ij} \right|^2 \right|^2 \right), i \in \mathcal{O}', j \in \mathcal{J}. \end{aligned} \tag{26}$$

Therefore, the problem $\mathcal{P}1.4$ can be transformed into its dual problem as

$$\begin{aligned} \mathcal{P}1.5: \quad & \min_{\boldsymbol{m}_j} \quad \sum\nolimits_{i\in\mathcal{O}'} \boldsymbol{m}_j^H \boldsymbol{m}_j \\ \text{s.t.} \quad & \\ & C11\text{–}C13 \end{aligned} \tag{27}$$

Algorithm 2 Adaptive Transmission Scheme.

Inputs: $\mathcal{O}, p_i^{tr}$
Repeat
Update $\mathcal{O}$
Solve $\mathcal{P}1.3$
  Obtain the feasible solution p_i^{tr}
  For UE $i \in O$
  If $p_i^{tr} < p_i^{tr*}$
    U_i is moved to O^H
  Else
    U_i is moved to $\mathcal{R}$
  End if
  Update $O^H \leftarrow \mathcal{O}$
Update p_i^{tr*}
If p_i^{tr} satisfy $C6$ and $C7$
  Update p_i^{tr}
Else
  If $\sum_{i\in\mathcal{O}} f_j > F_{j,\max}^C$
    Order $T_{i,\max} - T_i^o$
    Find $i^* = \arg\max_i \left(T_{i,\max} - T_i^o\right)$
    Move UE i^* to $\mathcal{R}$
Until $\sum_{i\in O} f_j - F_{j,\max}^C = \min\left(0, \left[\sum_{i\in O} f_j - F_{j,\max}^C\right]^+\right)$
Solve $\mathcal{P}1.5$
  Obtain $v_j, \mathcal{O}'$
    If $\mathcal{O}' \cup \mathcal{O} \neq \emptyset$
      Reassign $\left(B_{j,\max} - \sum_{i\in O'} B_{ij}, 0\right)^+$ to UE $i \notin \mathcal{O}$ obtained in Algorithm 1
      $i = \arg\max_i x_i$
      Until $B_{j,max} - \sum_{i\in\mathcal{O}'} B_{ij} = \min\left(0, B_{ij}\right)^+, i \in \mathcal{O}'$
    End if
Update $\mathcal{O}^H$

Update $\mathcal{O}^H$ **and** p_i^{tr*}

As proved that $\mathcal{P}1.5$ is a convex problem and can be solved efficiently, i.e., using the interior point method as proved in [29]. $\mathcal{P}1.4$ can take the same optimal value with the same set of beamforming vectors, i.e., $\boldsymbol{m}_j$ can be set to be identical. It is not difficult to obtain the feasible solution of p_i^{tr} to the problem $\mathcal{P}1.3$. Afterward, only when the p_i^{tr} meets the constraints of $C6$ and $C7$, UE i will consider to offload its task to MEC. Moreover, for the updated task offloading set $\mathcal{O}$, tasks with minimal remaining time, i.e., $i^* = \arg\max_i \left(T_{i,\max} - T_i^o\right)$, will be executed in priority. The channel bandwidth resources will be reassigned according to Algorithm 2. In such a case, one can save the bandwidth resource waste and more UEs are capable of offloading tasks to MEC for further execution. A proposed low-complexity heuristic algorithm that can obtain a near-optimal solution in a fast manner, which is described as Algorithm 2. One should note that $\mathcal{O}'$ mentioned in $\mathcal{P}1.3$ and $\mathcal{P}1.5$ may not be the same as shown in $\mathcal{P}1.3$ since some tasks may be dropped by MEC due to the limitation of communication and computation resources.

4. Performance evaluation

In this section, performance evaluation is presented to show the effectiveness of the proposed scheme. An example of the simulation environment is shown in Fig. 3, where there are $N = 75$ single-antenna UEs and one MEC. All UEs are assumed to be randomly distributed in a square area of coordinates $[0, 1000] \times [0, 1000]$ m. It is assumed that the small-scale fading is an independent Gaussian distributed as $\mathcal{CN}(0, 1)$. The offloading requests are received by RRHs and forward to BBU pool for execution. After receiving the instructions via downlink, UEs will dynamically adjust the offloading decisions and the channel allocation

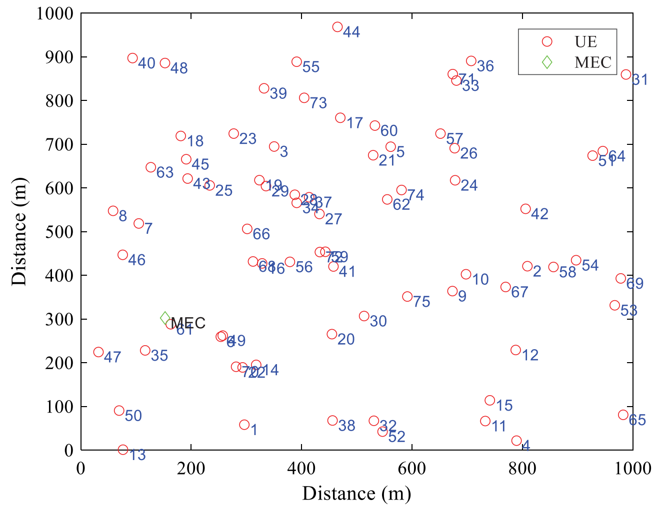

Fig. 3. The simulation environment, for example where we utilize $N = 75$ UEs and one MEC assumed to be randomly distributed in a square area with $[0, 1000] \times [0, 1000]$ m.

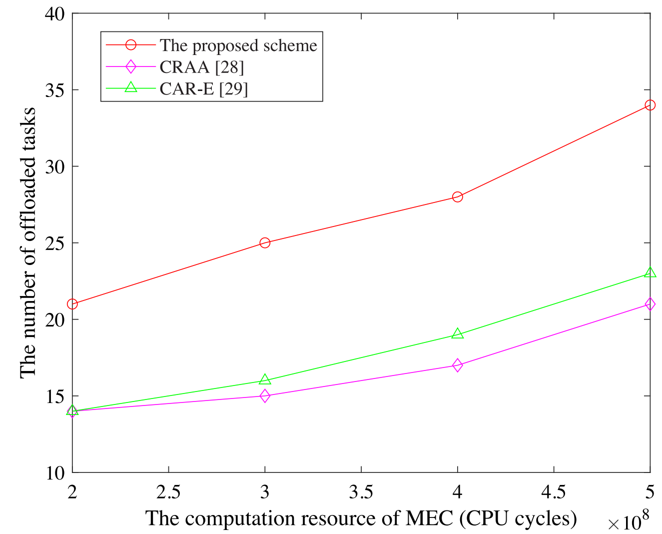

Fig. 5. The total number of offloaded tasks versus computation resources of MEC when $N = 75$ UEs.

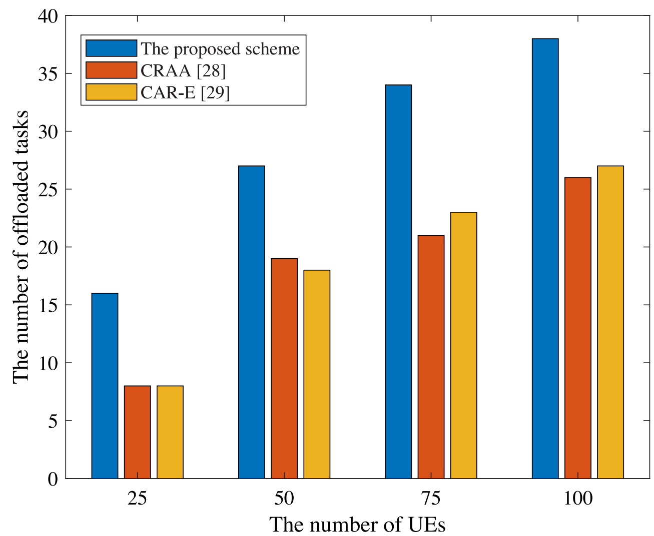

Fig. 4. The number of offloading requests versus the number of UEs.

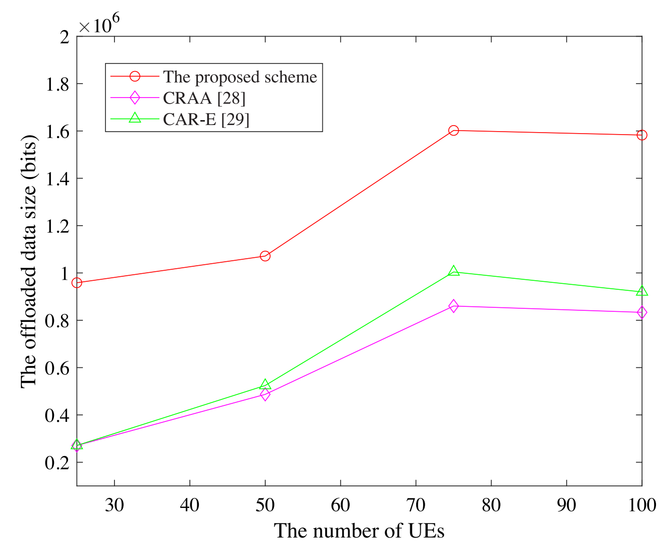

Fig. 6. The number of offloaded tasks in bits versus the number of UEs under 20 MHz.

strategy. The network bandwidth is set as 20 MHz and the maximum power for all UEs is set to 1 W. The tasks are generated randomly. Moreover, some selected significant parameters are obtained and compared with two novel low complexity algorithms, i.e., CRAA [29] and CAR-E [30]. The performance evaluation was implemented using MATLAB in the 64-bit Windows 10 Professional operating system with an Intel Xeon(R) E5-1630@3.70 GHz CPU processor and 32 GB RAM. The detailed explanations of some significant selected results are given in detail below.

We first investigate the relationship between the number of offloaded tasks and the number of UEs under different schemes. Fig. 4 illustrates the number of UEs versus the number of offloading requests under numerous typical values of UEs when $f_j = 5 * 10^8$ CPU cycles. One can see that with the number of UE increases from 25 to 100, the offloaded tasks increasing corresponding and the proposed scheme achieves the higher number of successfully offloaded tasks than CRAA and CAR-E schemes. When $N = 25$, the performance of CRAA and CAR-E is almost the same because the MEC can offer sufficient computation resources to serve UEs. With a higher number of UEs, e.g., $N = 75$ UEs and 100 UEs, the CAR-E achieves better performance than CRAA regarding the number of offloaded tasks. We select $N = 75$ UEs for further investigation. The proposed scheme reaches about 35 offloaded tasks while CRAA and CAR-E achieve around 21 and 23, respectively. Fig. 5 presents the relationship between the number of offloading requests and MEC's computation capacity. One can expect that higher computation capacity promises a higher number of offloaded tasks. The proposed scheme achieves the best performance among three different techniques. When $f_j = 2 * 10^8$ CPU cycles, the offloaded tasks are 18, 13 and 12 for the proposed scheme, CRAA and CAR-E, respectively. With the increase of the computation resources of MEC, when $f_j = 4 * 10^8$ CPU cycles, the number of offloaded tasks increases to nearly 27 while this value is about 18 and 16 for the CAR-E and CRAA, respectively.

Fig. 6 demonstrates the relationship between the number of offloaded tasks in bits and the number of UEs when the computation resource of MEC is set as $5 * 10^8$ CPU cycles. one can see that when $N = 25$ UEs, the offloaded tasks are around $0.93 * 10^6$ bits for the proposed scheme while this number is approximately $0.27 * 10^6$ bits for CRAA and CAR-E schemes. Moreover, since CAR-E scheme promise priority offloading tasks with the higher value of D_i, one can see the number

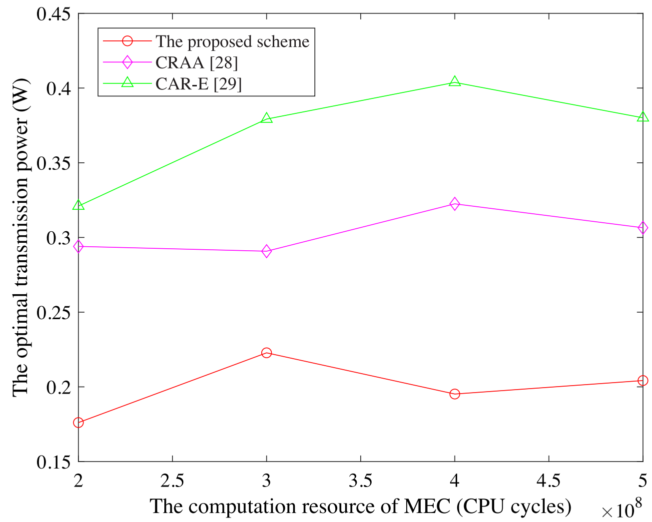

Fig. 7. The transmission power versus MEC computation resource when N = 75 UEs.

of CAR-E achieves smaller number of offloaded tasks than CRAA as shown in Fig. 4 even though CAR-E can offload a higher number of tasks in bits. With a higher number of UEs, e.g., when $N = 100$ UEs, the number of offloaded tasks in bits will decrease to some degrees for all three schemes. This is due to the reason that a larger number of UEs will occur severe interference and tasks with lower data size and non-strict latency requirements are more likely to achieve successful task offloading.

Fig. 7 presents the relationship between the computation capacity of MEC and the optimal transmission power. One can see that when MEC has the minimum computation capacity, e.g., when $f_j = 2 * 10^8$ CPU cycles, the optimal power is around 0.18 W for the proposed scheme and this value is approximately 0.29 W and 0.32 W for CRAA and CAR-E, respectively. With the computation capacity of MEC increase, the optimal power for the proposed scheme will increase from 0.14 W to 0.23 W where it obtains the highest value when $f_j = 3 * 10^8$ CPU cycles. This is due to the fact that when the computation resource of MEC cannot afford all offloading requests, UEs will increase the transmission power to guarantee the link quality. With the computation capacity of MEC increases, the available computation resource allocation to each offloaded task will increase, and thus the execution time cost as mentioned in Eq. (6) will decrease. As a result, when MEC has sufficient computation resources, the execution cost will decrease, and thus the offloading time as demonstrated in Eq. (5) can be prolonged hereby, which requires a lower transmission rate. Therefore, the transmission power will decrease. One should note that different characteristics of tasks, either different task size or CPU cycles required to execute the task, may influence network performance. Details regarding this aspect can be found in our previous work and some up-to-date Refs. [7,34,35]. Consider some exciting and emerging applications of MEC such as video caching in edge Cloud and task offloading for MEC in the ultra-dense network, the generated tasks may require enormous computation resources at MEC side due to the strict QoS requirements [36–38].

5. Conclusion and future work

In this paper, we analyze the network resource scheduling issue in depth by considering both UEs' experience and the mobile operator resource allocation, which is a key issue in 5G network services. We formulate the task scheduling and resource allocation problem as an NP-hard non-convex mixed-integer problem and then solved the original problem by utilizing a proposed heuristic scheme, which achieves a near-optimal solution with low-complexity. In this way, the original problem is transformed into a two-stage problem. First, we propose a hypergraph-based channel pre-allocation algorithm that can significantly decrease the mutual interference. Second, we consider UEs' offloading benefits by jointly consider computation resources allocation of MEC. Then, the offloading outcomes obtained by MEC will instruct all UEs to dynamically adjust the offloading decision via downlink. The results show that the proposed scheme achieves a higher number of offloaded tasks and offloaded tasks in bits than the existing techniques. Moreover, the relationship between the offloaded tasks in bits, the optimal transmission power and the computation resource of MEC is obtained.

The future work will focus on the new joint optimization framework of communication, computation and smart coded caching in MEC networks and its novel applications such as In-Edge AI [39]. Moreover, the co-located Cloud technique is a promising technique to realize the resource-hungry services for MEC [40], which still needs further investigation. Another interesting topic is that utilizing large scale reinforcement learning for the dynamic pricing strategy that can balance the economic cost of the mobile operator and the QoS of UEs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yangzhe Liao: Conceptualization, Methodology, Writing. **Liqing Shou:** Investigation, Writing. **Quan Yu:** Project administration, Formal analysis, Review & editing. **Qingsong Ai:** Validation, Review & editing. **Quan Liu:** Project administration, Supervision, Review & editing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under 51675389, 61801342 and the Natural Science Foundation of Hubei Province, China under Grant 2019CFB323.

References

[1] M. Chen, NDNC-BAN: supporting rich media healthcare services via named data networking in cloud-assisted wireless body area networks, Inform. Sci. 284 (2014) 142–156.
[2] Q.D. La, M.V. Ngo, T.Q. Dinh, T.Q. Quek, H. Shin, Enabling intelligence in fog computing to achieve energy and latency reduction, Digit. Commun. Netw. 5 (1) (2019) 3–9.
[3] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656.
[4] Y. Wang, R. Chen, D.-C. Wang, A survey of mobile cloud computing applications: Perspectives and challenges, Wirel. Pers. Commun. 80 (4) (2015) 1607–1623.
[5] M. Chen, Y. Zhang, Y. Li, S. Mao, V.C. Leung, EMC: Emotion-aware mobile cloud computing in 5G, IEEE Netw. 29 (2) (2015) 32–38.
[6] Z. Kuang, Y. Shi, S. Guo, J. Dan, B. Xiao, Multi-user offloading game strategy in OFDM mobile cloud computing system, IEEE Trans. Veh. Technol. (2019).
[7] Y. Liao, Q. Yu, Y. Han, M. Leeson, Relay-enabled task offloading management for wireless body area networks, Appl. Sci. 8 (8) (2018) 1409.
[8] A. Alnoman, G.H. Carvalho, A. Anpalagan, I. Woungang, Energy efficiency on fully cloudified mobile networks: Survey, challenges, and open issues, IEEE Commun. Surv. Tutor. 20 (2) (2017) 1271–1291.
[9] K. Wang, K. Yang, Resource allocation between service computing and communication computing for mobile operator, in: 2017 IEEE International Conference on Communications, ICC, IEEE, 2017, pp. 1–6.
[10] X. Cao, F. Wang, J. Xu, R. Zhang, S. Cui, Joint computation and communication cooperation for mobile edge computing, in: 2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt, IEEE, 2018, pp. 1–6.
[11] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2322–2358.
[12] B.P. Rimal, D.P. Van, M. Maier, Mobile edge computing empowered fiber-wireless access networks in the 5G era, IEEE Commun. Mag. 55 (2) (2017) 192–200.

[13] B. Varghese, R. Buyya, Next generation cloud computing: New trends and research directions, Future Gener. Comput. Syst. 79 (2018) 849–861.
[14] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, P. Mohapatra, Edge cloud offloading algorithms: Issues, methods, and perspectives, ACM Comput. Surv. 52 (2019) 2.
[15] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, R.P. Liu, Energy-efficient admission of delay-sensitive tasks for mobile edge computing, IEEE Trans. Commun. 66 (6) (2018) 2603–2616.
[16] J. Liu, Y. Mao, J. Zhang, K.B. Letaief, Delay-optimal computation task scheduling for mobile-edge computing systems, in: 2016 IEEE International Symposium on Information Theory, ISIT, IEEE, 2016, pp. 1451–1455.
[17] J. Li, H. Gao, T. Lv, Y. Lu, Deep reinforcement learning based computation offloading and resource allocation for MEC, in: 2018 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2018, pp. 1–6.
[18] S. Meng, Y. Wang, Z. Miao, K. Sun, Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment, Peer-to-Peer Netw. Appl. 11 (3) (2018) 462–472.
[19] T.X. Tran, D. Pompili, Joint task offloading and resource allocation for multi-server mobile-edge computing networks, IEEE Trans. Veh. Technol. 68 (1) (2018) 856–868.
[20] X. Zhang, Y. Mao, J. Zhang, K.B. Letaief, Multi-objective resource allocation for mobile edge computing systems, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, IEEE, 2017, pp. 1–5.
[21] W. Ni, H. Tian, X. Lyu, S. Fan, Service-dependent task offloading for multiuser mobile edge computing system, Electron. Lett. 55 (15) (2019) 839–841.
[22] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks, IEEE Access 4 (2016) 5896–5907.
[23] M. Merluzzi, P. Di Lorenzo, S. Barbarossa, V. Frascolla, Joint resource allocation for latency-constrained dynamic computation offloading with MEC, in: 2019 IEEE Wireless Communications and Networking Conference Workshop, WCNCW, IEEE, 2019, pp. 1–6.
[24] C.-M. Huang, M.-S. Chiang, D.-T. Dao, W.-L. Su, S. Xu, H. Zhou, V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture, IEEE Access 6 (2018) 17741–17755.
[25] S. Huang, A. Elgammal, D. Yang, On the effect of hyperedge weights on hypergraph learning, Image Vis. Comput. 57 (2017) 89–101.
[26] L. Wang, H. Wu, Y. Ding, W. Chen, H.V. Poor, Hypergraph-based wireless distributed storage optimization for cellular D2D underlays, IEEE J. Sel. Areas Commun. 34 (10) (2016) 2650–2666.
[27] H. Zhang, L. Song, Y. Li, G.Y. Li, Hypergraph theory: Applications in 5G heterogeneous ultra-dense networks, IEEE Commun. Mag. 55 (12) (2017) 70–76.
[28] K. Wang, K. Yang, C. Pan, J. Wang, Joint offloading framework to support communication and computation cooperation, CoRR (2017).
[29] K. Wang, P.-Q. Huang, K. Yang, C. Pan, J. Wang, Unified offloading decision making and resource allocation in ME-RAN, IEEE Trans. Veh. Technol. 68 (8) (2019) 8159–8172.
[30] M.L. Wang, S. Hosseinalipour, Z. Gao, Y. Tang, L. Huang, H. Dai, Allocation of computation-intensive graph jobs over vehicular clouds in IoV, IEEE Internet Things J. (2019).
[31] Q. Li, G. Kim, R. Negi, Maximal scheduling in a hypergraph model for wireless networks, in: 2008 IEEE International Conference on Communications, 2008, pp. 3853–3857.
[32] Q. Li, G. Kim, R. Negi, Maximal scheduling in a hypergraph model for wireless networks, in: 2008 IEEE International Conference on Communications, 2008, pp. 3853–3857.
[33] A. Wiesel, Y.C. Eldar, S. Shamai, Linear precoding via conic optimization for fixed MIMO receivers, IEEE Trans. Signal Process. 54 (1) (2005) 161–176.
[34] Y. Liao, Y. Han, Q. Yu, Q. Ai, Q. Liu, M.S. Leeson, Wireless body area network mobility-aware task offloading scheme, IEEE Access 6 (2018) 61366–61376.
[35] R. Chai, J. Lin, M. Chen, Q. Chen, Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems, IEEE Syst. J. 13 (4) (2019) 4110–4121.
[36] M. Chen, Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network, IEEE J. Sel. Areas Commun. 36 (3) (2018) 587–597.
[37] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, I. Humar, A dynamic service migration mechanism in edge cognitive computing, ACM Trans. Internet Technol. 19 (2) (2019) 30.
[38] Y. Hao, L. Hu, Y. Qian, M. Chen, Profit maximization for video caching and processing in edge cloud, IEEE J. Sel. Areas Commun. 37 (7) (2019) 1632–1641.
[39] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, M. Chen, In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning, IEEE Netw. 33 (5) (2019) 156–165.
[40] M. Chen, Y. Hao, C.F. Lai, D. Wu, Y. Li, K. Hwang, Opportunistic task scheduling over co-located clouds in mobile environment, IEEE Trans. Serv. Comput. 11 (3) (2016) 549–561.