IET Communications

Research Article

Computation offloading and resource allocation for mobile edge computing with multiple access points

ISSN 1751-8628
Received on 23rd April 2019
Revised 16th June 2019
Accepted on 24th June 2019
E-First on 26th September 2019
doi: 10.1049/iet-com.2019.0446
www.ietdl.org

Qiuping Li[1,2], *Junhui Zhao*[1,3] ✉, *Yi Gong*[2]

[1]*School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, People's Republic of China*
[2]*Shenzhen Engineering Laboratory of Intelligent Information Processing for IoT, Southern University of Science and Technology, Shenzhen 518055, People's Republic of China*
[3]*School of Information Engineering, East China Jiaotong University, Nanchang 330013, People's Republic of China*
✉ *E-mail: junhuizhao@hotmail.com*

Abstract: Mobile edge computing (MEC) is an innovative computing paradigm to enhance the computing capacity of mobile devices (MDs) by offloading computation-intensive tasks to MEC servers. With the widespread deployment of wireless local area networks, each MD can offload computation task to server via multiple wireless access points (WAPs). However, computation offloading can bring a higher system cost if all users select the same access points to offload their tasks. This study proposes a computation offloading strategy and resource allocation optimisation scheme in a multiple wireless access points network with MEC, which aims to minimise the system cost by providing the optimal computation offloading strategy, transmission power allocation, bandwidth assignment, and computation resource scheduling. The proposed scheme decouples the optimisation problem into subproblems of offloading strategy and resource allocation since the problem is NP-hard. The offloading strategy involves the optimal access point selection, which is analysed by the potential game. The resource allocation is obtained using Lagrange multiplier. The authors' analysis and simulation results verify the convergence performance of the proposed scheme, and the proposed scheme outperforms the simple resource allocation scheme and the offloading strategy optimisation scheme in terms of the system cost.

1 Introduction

With the rapid development of the Internet of Things and wireless technologies, highly advanced mobile applications, such as augmented reality face recognition and speech recognition, are emerging and attracting [1–3]. These applications typically result in high energy consumption and require powerful computation capacity [4]. However, mobile devices (MDs) are usually limited in computation capabilities and battery power [5, 6]. The limited computing resources and battery power of MDs have become a significant constraint for future mobile applications development. Cloud computing with enormous computation and storage resources has been successfully developed to relieve the conflict between resource-intensive applications and resource-constrained MDs [7, 8], where MDs can offload their computation tasks to the remote cloud data centres [9]. Moreover, in the research of cloud computing, some remarkable studies have been conducted, which mainly concentrated on improving the performance in terms of energy and cost [10–13].

Although cloud computing can enhance the computing capacity of MDs, the long-distance data transmission can lead to unacceptable latency and extra transmission energy consumption when the MDs delivery the tasks to the remote cloud servers [14]. To further tackle the above challenges, mobile edge computing (MEC), which is extending computing and storage resources in close proximity to mobile users, has been proposed as a supplement to cloud computing [15]. MEC avoids transmitting massive computing tasks to the remote cloud servers. Thus, MEC is expected to process these computation tasks faster, and save more energy than clouds. In this paper, we consider a MEC system and investigate a problem of computation offloading and resource allocation.

1.1 Related work

Many studies involved in the MEC computation offloading have recently attracted much attention in the academia and industry. The main focus of investigators is on improving the system performance gain, such as reducing the system cost or energy consumption, by optimising the offloading decisions and allocating resource effectively (e.g. the allocation of computation resource, bandwidth, and transmission power).

The authors investigated cooperation among fog nodes to achieve minimal latency in [16] and [17]. Tao *et al.* proposed an energy minimising optimisation problem while guaranteeing the quality of service of users [18]. The authors in [19] designed the optimal policy for controlling offloading data and time/sub-channel allocation to minimise the weighted sum mobile energy consumption. In [20], a distributed computation offloading algorithm was proposed in a multi-channel wireless interference environment, which minimised the system energy consumption and delay by finding the optimal computation offloading decision. The authors in [21] recommended a steerable economic expense algorithm to minimise the average expenses of users in a device-to-device-enabled MEC system. In [22], the system delay and cost were minimised by utilising queueing and convex optimisation theories. By jointly optimising the offloading decisions of all users as well as the allocation of computation and communication resources, the authors in [23] minimised the overall cost of energy, computation, and delay for all users, Du *et al.* in [24] minimised the maximum cost among users to ensure the fairness of all mobile users, the total energy consumption of all users was minimised in [25] .

Unfortunately, the studies in [18–20] did not optimise the transmission power and computation resource allocation, and the works in [23, 25] disregarded the optimal transmission power assignment. Meanwhile, aforementioned studies mainly investigated computation offloading problem from the perspective a single wireless access point (WAP) [18–25]. Actually, wireless local area networks has been widely deployed, where each MD usually connects to the Internet via more than one WAP, e.g. heterogeneous networks. Moreover, the ETSI has renamed the MEC to multi-access edge computing to support multiple access technologies. For a multiple WAPs system, if many MDs choose

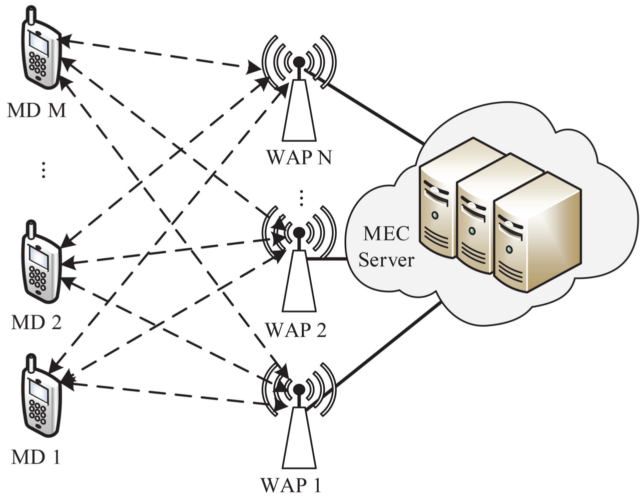

Fig. 1 *System model*

the same WAP to offload their tasks to the MEC servers, computation offloading can increase the task processing delay and cause high system cost. Therefore, it is necessary to design a novel computation offloading scheme for computation offloading in a multiple WAPs network with MEC, where the load of WAPs should be balanced.

Recently a few works noted the multiple WAPs networks in computation offloading [26–29]. The work in [26] analysed the computation offloading problem of multiuser MEC system based on the game-theoretic, where each user could offload its computation task to a common edge-server by choosing a proper WAP among multiple WAPs. Considering the overlapping networks, the authors in [27] minimised the total cost of all users by evolutionary game. In [28], the authors derived the optimal computation distribution among clouds and optimal computation offloading strategy in heterogenous networks. Zhang *et al.* presented an energy-efficient computation offloading scheme in 5G heterogeneous networks [29], where the energy consumption of users was minimised by jointly optimising the offloading strategy and radio resource allocation.

However, the works in [26–29] did not consider optimising the offloading strategy and resource allocation simultaneously. Similarly to the works in [26, 27] only investigated computation offloading strategy, without considering the optimal transmission power, bandwidth and computation resource allocation. The authors in [28] assumed only one user to offload computation task and the computation resource allocation of MEC server was not discussed. The work in [29] ignored the allocation of computation resource. The limited computing resources of the MEC server, and communication resources between users and WAPs are shared by multiple users, where computation and communication resources should be jointly allocated for further improving system performance gain.

1.2 Contribution and organisation

On the one hand, notwithstanding some studies related to MEC computation offloading jointly considered offloading decision making and resource allocation [18–25], these works generally investigated computation offloading problem under the setting of a single WAP, and it did not consider the overlapping networks. In reality, there may exist the case that multiple WAPs connect to a common MEC server. In a multi-WAP network with MEC, without coordination among WAPs, the performance gain and offloading efficiency can be limited. On the other hand, even though some previous studies sought to investigate the offloading policies in a MEC system with multiple WAPs [26–29], none of them jointly considered how to schedule computation tasks and optimise resource allocation. In this paper, we aim to solve the computation offloading strategy problem of multiple MDs and efficiently optimise the computation resource allocation, transmit power assignment, and radio bandwidth allocation in a multiple WAPs network with MEC. According to the resource allocation, each MD not only decides whether offloading but also needs to choose an optimal WAP among multiple WAPs for offloading. The contributions of this paper are summarised as follows.

- In this paper, we consider the case that a MEC system has multiple WAPs to provide data transmission service, and the computation offloading and resource allocation problem for this case can be investigated.
- This paper then formulates the computation offloading problem of multiple MDs connecting multiple WAPs as a system cost minimisation problem while guaranteeing the computation execution time constraint of MDs, where the WAP selection is formulated as an offloading decision. The considered problem is difficult to solve due to the combinatorial nature of multi-MD computation offloading decisions and the complexity of the optimisation objective. We hence design a computation offloading strategy and resource allocation optimisation (CSAO) scheme to solve it.
- The CSAO scheme is proposed to decouple the computation offloading problem into two subproblems: (i) the offloading strategy, which involves the optimal WAP selection, and (ii) joint resource allocation for the offloading MDs, which can be further decomposed into the problems of computation resource allocation and radio resource allocation (transmission power and bandwidth allocation). We use the potential game for computation offloading strategy, which balances the load among multiple WAPs. The radio resource allocation is found using a bisection method, and the computation resource allocation is achieved applying Lagrange multiplier. The proposed CSAO scheme works iteratively between offloading decisions and resource allocation to minimising system cost. Numerical results demonstrate that our proposed CSAO scheme can effectively drop the system cost in comparison with the existing schemes.

The remainder of this paper is structured as follows. The system model and system cost minimisation problem are presented in Section 2. The computation offloading strategy and resource allocation optimisation scheme is described in Section 3. Simulation results are shown in Section 4. Finally, Section 5 concludes the entire work.

2 System model, theoretical model, problem formulation

2.1 System model

The system model is shown in Fig. 1, where we consider a MEC server with total computing resource F, I WAPs and M MDs. The set of WAPs and MDs in the network are denoted as $\mathscr{I} = \{1, 2, \ldots, I\}$ and $\mathscr{M} = \{1, 2, \ldots, M\}$, respectively. Each MD has a computation task and computation task of MD m can be modelled as a 3-tuple $\varphi_m = \{C_m, D_m, t_m^{\max}\}$, where C_m denotes the required computing resource to complete the computation task φ_m, D_m is data size of the computation task φ_m and $t_m^{\max}$ is the tolerable maximum latency for accomplishing the task. All WAPs are connected to a common MEC server, and maximal available bandwidth of WAP i is B^i.

Each MD can offload its computation task to a common MEC server by choosing a proper WAP in a set of WAPs. For each MD, the task cannot be divided and has to be processed as whole either locally at the MD or offloaded to MEC server. Then, we define the computation offloading strategy of MDs as $\mathscr{A} = \{a_m | a_m = i, m \in \mathscr{M}, i \in \Phi\}$, where $\Phi = \{0, 1, 2, \ldots, I\}$ is the offloading decision set and $a_m (a_m \in \Phi)$ is the offloading decision of MD m. Specially, $a_m = i (i \in \mathscr{I})$ means that MD m offloads the computation task to the MEC server through WAP i, and $a_m = 0$ represents that MD decides to execute the task locally. Moreover, for many applications, e.g. natural language processing, the application offloading can be completed during a time shorter than the timescales of MD mobility and the dynamics of wireless networks. Hence, similar to the existing studies [18, 20, 24, 27], to enable tractable analysis, a quasi-static scenario is considered in this paper, where we assume that all MDs and wireless networks remain stationary during a computation offloading period.

In the proposed system model, the WAP selection is integrated into the computation offloading strategy. The load of WAPs can be balanced by optimising the computation offloading decisions. Moreover, in the following, based on the proposed system model, we analyse the cost for the task in local processing and MEC processing.

2.2 Cost under different computation models

In the sub-section, the task processing delay is discussed, and the total cost under different computation decisions is analysed from the perspective of energy consumption and monetary cost.

2.2.1 MEC processing model.:

(a) *MEC Processing Delay:* If MD m decides to process its computation task φ_m in MEC, the processing delay includes three parts. The first one is the transmission time that MD m transmits the input data to the MEC server through WAP i. The second part is the computation time that MEC processes the computation task φ_m. The third part is transmission time that the computation results back to MD m. Moreover, like any other studies [15, 29], the time of the third part will is negligible, as the amount of output data is much smaller than the input data.

When MD m transmits the input data to the MEC server through WAP i, we allocate orthogonally a partition of maximal available bandwidth of WAP i to MD m [30]. Denote the channel gain between MD m and WAP i as g_m^i, then available transmission rate of MD m at WAP i can be formulated as

$$r_m^i = B_m^i \log_2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right), \tag{1}$$

where B_m^i is the bandwidth that MD m is allocated at WAP i, p_m^i is the transmission power of MD m at WAP i, and δ_i^2 is the noise power.

Then, the task processing delay for offloading computation task φ_m from MD m to MEC server via WAP i can be expressed as

$$t_m^i = \frac{C_m}{f_m^i} + \frac{D_m}{r_m^i}, \tag{2}$$

where f_m^i is the computation resource allocated to MD m when MD m offloads its computation task via WAP i.

(b) *Offloading Energy Consumption:* The energy consumption of MD m in MEC processing model involves scanning energy consumption, transmission energy consumption and maintain energy consumption, which is given by

$$E_m^i = E_m^s + p_m^i \frac{D_m}{r_m^i} + p_m^c \frac{C_m}{f_m^i}, \tag{3}$$

where E_m^s denotes the energy consumed when MD m scans available WAPs [26] and p_m^c denotes the maintaining power of MD m in idle state. Furthermore, p_m^c is constant in this paper.

(c) *Monetary Cost:* Similar to the existing studies [4, 22], each MD with using the computing resource of MEC needs to pay for the service, and it is expressed as

$$M_m^i = p f_m^i, \tag{4}$$

where p denotes the unit cost for MEC computation resource.

(d) *MEC Processing Cost:* According to (3) and (4), for the case that m offloads its computation task to MEC through WAP i, the total cost of MEC processing for MD m is defined as

$$z_m^i = e_m E_m^i + c_m M_m^i, \tag{5}$$

where e_m and c_m are the impact factor of energy consumption and monetary cost, respectively.

2.2.2 Local processing model.:

The computational time and the energy consumption of computation task φ_m by computing locally can be respectively given by [23, 26]

$$t_m^0 = \frac{C_m}{f_m^0}, \tag{6}$$

$$E_m^0 = k\left(f_m^0\right)^2 C_m, \tag{7}$$

where f_m^0 is the computation capability of MD m and k is the effective switched capacitance relying on the chip architecture.

Compared to MEC processing model, each MD uses its computing resource to process the computation task in the local processing model, where the MDs do not need to pay for the service. Therefore, in this case, the total cost of MD m is equal to its energy consumption, and we have $z_m^0 = E_m^0$.

In this sub-section, we define the total cost under different computation decisions based on the two metrics such as energy consumption, and cost of computation resource. It is taken by a MD as a result of its computation offloading.

2.3 Problem formulation

In this sub-section, the computation offloading optimisation problem of multiple MDs connecting multiple WAPs is formulated. The objective is to minimise the system cost by providing the optimal offloading strategy $\mathscr{A}$, bandwidth allocation $\boldsymbol{B}^*$, transmission power $\boldsymbol{P}^*$ and computation resource allocation $\boldsymbol{f}^*$. The system cost is defined as $z(\mathscr{A}, \boldsymbol{f}, \boldsymbol{B}, \boldsymbol{P}) = \sum_{m \in M} z_m(\mathscr{A}, \boldsymbol{f}, \boldsymbol{B}, \boldsymbol{P})$, where $z_m = (1 - \sum_{i \in I} K(a_m = i)) z_m^0 + \sum_{i \in I} K(a_m = i) z_m^i$, $\forall m \in \mathscr{M}$, $K(x)$ is an indicator function, and $K(x) = 1$ if x is true, otherwise $K(x) = 0$. The optimisation problem of joint computation offloading decision and resource allocation is formulated as

$$\begin{aligned}
\min_{\boldsymbol{B}, \boldsymbol{P}, \boldsymbol{A}, \boldsymbol{f}} \quad & z(\mathscr{A}, \boldsymbol{f}, \boldsymbol{B}, \boldsymbol{P}) = \sum_{m=1}^{M} z_m(\mathscr{A}, \boldsymbol{f}, \boldsymbol{B}, \boldsymbol{P}), \\
\text{s.t.} \quad & C1: f_m^0 \geq 0, \quad \forall m \in \mathscr{M}, \\
& C2: t_m^i \leq t_m^{\max}, \quad \forall m \in \mathscr{M}, \forall i \in \Phi, \\
& C3: 0 \leq p_m^i \leq p_m^{\max}, \quad \forall m \in \mathscr{M}, \forall i \in \mathscr{I}, \\
& C4: \sum_{m \in \mathscr{M}} B_m^i \leq B^i, \quad \forall m \in \mathscr{M}, \forall i \in \mathscr{I}, \\
& C5: \sum_{i \in \mathscr{I}} \sum_{m \in \mathscr{M}} f_m^i \leq F, \quad \forall m \in \mathscr{M}, \forall i \in \mathscr{I}, \\
& C6: a_m = \{0, 1, 2, \ldots, I\}, \quad \forall m \in \mathscr{M}, \\
& C7: \sum_{i \in \Phi} K(a_m = i) = 1, \quad \forall m \in \mathscr{M}, \forall i \in \Phi,
\end{aligned} \tag{8}$$

where B^i is the total bandwidth of WAP i. Constraint $C1$ represents the available computing resource of MD which is non-negative. Constraint $C2$ states that each computation task should be processed before a tolerable deadline. Constraint $C3$ makes sure that the transmit power of MD m does not exceed the maximum transmit power $p_m^{\max}$. For constraint $C4$, it states that the total bandwidth occupied by the MDs is less than the overall bandwidth of WAP i ($i \in \mathscr{I}$). Constraint $C5$ ensures that the total computing resource assigned to offloading MDs do not exceed the maximum computing resource of the MEC server. According to constraints $C6$ and $C7$, every task is executed either locally or remotely, and every offloading MD only chooses at most one WAP to transmit its task.

The problem in (8) is a mixed-integer and non-linear optimisation problem, since the computation offloading strategy $\mathscr{A}$

is an integer variable and resource allocation vectors (bandwidth allocation $\boldsymbol{B}$, transmission power assignment $\boldsymbol{P}$ and computation resource allocation $\boldsymbol{f}$) are continuous. Moreover, the optimisation variables are highly complex coupling. As a result, it is quite intractable to solve (8) by directly applying the standard optimisation techniques. Hence, in order to address problem (8), a CSAO scheme is presented in the next section, which decomposes problem (8) into subproblems of computation offloading strategy and resource allocation.

3 Computation offloading strategy and resource allocation optimisation scheme

It is difficult to obtain an optimal solution to the original optimisation problem (8). From the considered problem (8), it can be observed that the resource allocation constraints $C1$–$C5$ are decoupled from the computation offloading strategy constraints $C6$ and $C7$. Therefore, in this section, a CSAO scheme is proposed to decouple the original optimisation problem into two subproblems for obtaining the minimal system cost in a multi-WAP scenario. They are the offloading decision making and resource allocation (computing resource allocation, bandwidth allocation, and transmission power assignment), respectively. Concretely, we determine the offloading strategy under given computing and radio resources allocation, and then the optimal transmission power assignment, bandwidth allocation and computation resource allocation for the all MEC processing MDs can be determined when the computation offloading strategy is known. We iterate this process until convergence. Moreover, the offloading strategy and the resource allocation are detailed in Section 3.1 and Section 3.2, respectively. Finally, a joint algorithm for minimising system cost is designed and its algorithm complexity is analysed in Section 3.3.

3.1 Offloading decision

The game-theory can effectively solve the conflict of interest among the decision-making bodies to achieve the best strategy for optimal combination. Hence, in this sub-section, a computation offloading strategy game is proposed to analyse the current optimal offloading decision. In this game, the players are all MDs. The set of strategies for player m is $\mathscr{A}_m$. The cost function of MD m is denoted as $z_m(a_m, a_{-m})$, where $a_{-m} = (a_1, \ldots, a_{m-1}, a_{m+1}, \ldots, a_M)$ is the offloading decisions of all MDs apart from MD m. Accordingly, this game can be defined as $\Psi = \left\{\mathscr{M}, (\mathscr{A}_m)_{m \in \mathscr{M}}, \left(z_m^i\right)_{m \in \mathscr{M}, i \in \mathscr{I}}\right\}$.

Theorem 1: Computation offloading game Ψ always converges to a Nash Equilibrium (NE) and has the finite improvement property (FIP).

Proof: A game is called an exact potential game if it admits potential function $P(\boldsymbol{a})$ for every MD when the offloading strategy unilaterally change from a_m to a_m', and $a_{-m} \in \prod_{j \neq m} \mathscr{A}_j$, $a_m, a_m' \in \mathscr{A}_m$, we have $z_m(a_m, a_{-m}) - z_m(a_m', a_{-m}) = P(a_m, a_{-m}) - P(a_m', a_{-m})$. Moreover, every exact potential game with finite strategy sets always has a NE and the FIP [26]. Therefore, we first construct a potential function $P(\boldsymbol{a})$ [4], which is formulated as

$$
\begin{aligned}
P(\boldsymbol{a}) = \sum_{i=1}^{I} K(a_m = i)\{ e_m\left(p_m^i \frac{D_m}{r_m^i} + p_m^c \frac{C_m}{f_m^i}\right) + c_m M_m^i \\
+ \sum_{n=1, n \neq m}^{M} \left(k\left(f_n^0\right)^2 C_n - e_n E_n^s\right)\} + \sum_{n=1}^{M} \left(E_n^0 - e_n E_n^s\right) K(a_m = 0)\,.
\end{aligned}
\tag{9}
$$

Then, by considering the following two situations, it can be proven that the computation offloading strategy game Ψ is an exact potential game.

Case 1. MD m decides to offload its computation task via WAP j ($j \in \mathscr{I}$) or compute locally ($a_m = 0$). Based on (9), we know that

$$
\begin{aligned}
&P(0, a_{-m}) - P(j, a_{-m}) \\
&= k\left(f_m^0\right)^2 C_m - e_m E_m^s - e_m\left(p_m^j \frac{D_m}{r_m^j} + p_m^c \frac{C_m}{f_m^j}\right) - c_m q f_m^j \\
&= z_m(0, a_{-m}) - z_m(j, a_{-m})\,.
\end{aligned}
\tag{10}
$$

Case 2. MD m decides to offload the computation task via WAP i or j, and $i, j \in \mathscr{J}$. According to (9), we know that

$$
\begin{aligned}
&P(i, a_{-m}) - P(j, a_{-m}) \\
&= e_m\left(D_m\left(\frac{p_m^i}{r_m^i} - \frac{p_m^j}{r_m^j}\right) - p_m^c C_m\left(\frac{1}{f_m^i} - \frac{1}{f_m^j}\right)\right) + c_m q\left(f_m^i - f_m^j\right) \\
&= z_m(i, a_{-m}) - z_m(j, a_{-m})\,.
\end{aligned}
\tag{11}
$$

Based on the results in the two cases above, we can obtain the computation offloading game problem is an exact potential game, and hence always converges to a NE and has the FIP. When the game reaches a NE, no players have any intention to unilaterally break away from this steady state for sake of decreasing his cost. □

In the computing decision-making problem among multiple MDs, we utilise a potential game to update the computation offloading strategies until obtaining NE. After offloading strategies are given by all MDs, the computation and communication resource allocation needs optimisation to minimise the cost of all offloading MDs. The optimal resource allocation will be discussed in the next sub-section.

3.2 Resource allocation

After determining the computation offloading strategies, the optimal resource allocation should be performed for MDs that are required to offload the tasks to MEC. Based on (8), the optimisation problem of resource allocation is expressed as

$$
\begin{aligned}
\min \quad & \Theta(\boldsymbol{f}, \boldsymbol{B}, \boldsymbol{P}) = \sum_{i \in \mathscr{I}} \sum_{m \in \mathscr{M}_i} I(a_m = i) z_m^i, \\
\text{s.t.} \quad & C8: \frac{C_m}{f_m^i} + \frac{D_m}{r_m^i} \le t_m^{\max}, \quad \forall i \in \mathscr{I}, \forall m \in \mathscr{M}_t, \\
& C3, C4, C5, \quad \forall i \in \mathscr{I}, \forall m \in \mathscr{M}_i,
\end{aligned}
\tag{12}
$$

where $\mathscr{M}_t$ is a set of MEC processing MDs, and $\mathscr{M}_i$ denotes a set of MDs with offloading the computation task to MEC via WAP i ($i \in \mathscr{I}$), and $\sum_{i \in \mathscr{I}} |\mathscr{M}_i| = |\mathscr{M}_t|$. According to (12), the radio resource allocation (e.g. transmit power assignment and bandwidth allocation) and the computation resource allocation are decomposed both in the objective function and the constraints [24], which will be detailed in the following.

3.2.1 Problem of computation resource allocation.: The optimal computation resource allocation needs to be determined by solving the following optimisation problem:

$$
\begin{aligned}
\min_{f} \quad & \sum_{i \in \mathscr{I}} \sum_{m \in \mathscr{M}_i} e_m p_m^c \frac{C_m}{f_m^i} + c_m p f_m^i, \\
\text{s.t.} \quad & C5, C8 \quad \forall m \in \mathscr{M}_t, \forall i \in \mathscr{I}\,.
\end{aligned}
\tag{13}
$$

Derived from (14), we can know that $\partial^2 z_m^i / \partial\left(f_m^i\right)^2 > 0$ and constraint qualifications of (13) are linear. Therefore, problem (13) is a convex optimisation problem.

$$
\frac{\partial^2 z_m^i}{\partial\left(f_m^i\right)^2} = 2 e_m p_m^c \frac{C_m}{\left(f_m^i\right)^3}\,.
\tag{14}
$$

As problem (13) is convex, the slater condition is satisfied. Then, the partial Lagrange function is used to solve this problem, and it is expressed as

```
Initialization:
    Computation task $\varphi_m = \{C_m, D_m, t_m^{max}\}$, $m \in \mathcal{M}_t$, maximum number of iterations $v_{max}$ and precision $\varepsilon_1$.
while $v \le v_{max}$ do
    Let $\delta_s = 1/\left(10^4 * v\right)$, compute $f_m^i(v)$ according to substitute $\lambda$ and $\beta_m$ into (16), update Lagrange multiplier $\lambda$ and $\beta_m$ from (17) and (18) respectively.
    If$\|\lambda(v+1) - \lambda(v)\|_2 < \varepsilon_1$, $\|\beta(v+1) - \beta(v)\|_2 < \varepsilon_1$, let $f_m^{i^*} = f_m^i(t)$, and break the while-loop, otherwise $v = v + 1$.
end while
Output: $f_m^{i*}$, $m \in \mathcal{M}_t$.
```

Fig. 2 *Algorithm 1: Algorithm for Solving Problem (13)*

$$
\begin{aligned}
L(\boldsymbol{f}, \lambda, \boldsymbol{\beta}) = & \sum_{i \in \mathscr{J}} \sum_{m \in \mathscr{M}_i} \left(e_m p_m^c \frac{C_m}{f_m^i} + c_m p f_m^i \right) \\
& + \beta_m \left(\frac{C_m}{f_m^i} + \frac{D_m}{r_m^i} - t_m^{\max} \right) + \lambda \left(\sum_{i \in \mathscr{J}} \sum_{m \in \mathscr{M}_t} f_m^i - F \right).
\end{aligned} \tag{15}
$$

where $\lambda \ge 0$ and $\boldsymbol{\beta} \ge \boldsymbol{0}$ are the Lagrange multipliers related to $C5$ and $C8$, respectively.

Subsequently, the optimal computation resource allocation $f_m^{i^*}$ is obtained by the KKT condition. By differentiating $L(\boldsymbol{f}, \lambda, \boldsymbol{\beta})$ with respect to f_m^i, $m \in \mathscr{M}_t$, and makes it equal 0, it can be achieved that

$$
f_m^{i^*} = \sqrt{\frac{(e_m p_m^c + \beta_m^*) C_m}{c_m p + \lambda^*}}, \quad \forall m \in \mathscr{M}_t. \tag{16}
$$

Then, $f_m^{i^*}$ is iteratively derived by updating the Lagrange multipliers λ and β_m [4]. Moreover, the Lagrange multipliers are updated as follows

$$
\lambda(v+1) = [\lambda(v) + \delta_s(v) \nabla \lambda(v)]^+, \tag{17}
$$

$$
\beta_m(v+1) = [\beta_m(v) + \delta_s(v) \nabla \beta_m(v)]^+, \tag{18}
$$

where $[x]^+ = \max\{0, x\}$, v is the iteration index, $\delta_s(v)$ represents the step of the iteration, $\delta_s = 1/\left(10^4 * v\right)$, $\nabla \lambda = \sum_{m \in \mathscr{M}_t} f_m^i - F$, and $\nabla \beta_m = (C_m / f_m^i) + (D_m / r_m^i) - t_m^{\max}$. The entire procedure to solve problem (13) is described in Algorithm 1 (see Fig. 2).

3.2.2 Problem of transmission power and bandwidth allocation.: Under given computation resource allocation, the optimal transmission power and bandwidth allocation is formulated as

$$
\begin{aligned}
\min \quad & Y(\boldsymbol{B}, \boldsymbol{P}) = \sum_{i \in \mathscr{J}} \sum_{m \in \mathscr{M}_i} y_m^i, \\
\text{s.t.} \quad & C3, C4, C8, \quad \forall m \in \mathscr{M}_i, \forall i \in \mathscr{J}.
\end{aligned} \tag{19}
$$

where

$$
y_m^i = \frac{e_m p_m^i D_m}{B_m^i \log_2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)}.
$$

Theorem 2: The problem in (19) is convex.

Proof: In order to prove problem (19) is a convex optimisation problem, we first need to demonstrate the Hessian matrix of y_m^i is positive. If the Hessian matrix of y_m^i is positive definite, y_m^i is convex. Moreover, the Hessian matrix of y_m^i is shown as

$$
H\left(y_m^i\right) = \begin{bmatrix} \dfrac{\partial^2 y_m^i}{\partial \left(B_m^i\right)^2} & \dfrac{\partial^2 y_m^i}{\partial B_m^i \partial p_m^i} \\ \dfrac{\partial^2 y_m^i}{\partial p_m \partial B_m^i} & \dfrac{\partial^2 y_m^i}{\partial \left(p_m^i\right)^2} \end{bmatrix} \tag{20}
$$

In (20), the second-order derivatives of y_m^i with respect to bandwidth B_m^i is

$$
\frac{\partial^2 y_m^i}{\partial \left(B_m^i\right)^2} = \frac{2 S_m^i p_m^i}{\left(B_m^i\right)^3}, \tag{21}
$$

where

$$
S_m^i = \frac{e_m D_m}{\log_2(1 + x_m^i)} \quad \text{and} \quad x_m^i = \frac{p_m^i g_m^i}{\delta_i^2}.
$$

$S_m^i > 0$ as the value of SINR is larger than 1 ($x_m > 1$). Hence, we have $\partial^2 y_m^i / \partial \left(B_m^i\right)^2 \ge 0$.

The second-order derivatives of y_m^i with respect to transmission power p_m^i is

$$
\frac{\partial^2 y_m^i}{\partial (p_m^i)^2} = \frac{g_m^i S_m^i \Gamma_m^i}{\delta_i^2 B_m^i x_m^i} \left(2 \Gamma_m^i \left(1 + x_m^i\right) - \frac{2 + x_m^i}{1 + x_m^i} \right), \tag{22}
$$

where $\Gamma_m^i = x_m^i / \left(\left(\left(1 + x_m^i\right) \ln 2\right)^2\right)$ and $0 < \Gamma_m < 1$.

Additionally,

$$
\frac{\partial^2 y_m^i}{\partial B_m^i \partial p_m^i} = \frac{\partial^2 y_m^i}{\partial p_m^i \partial B_m^i},
$$

which is formulated as

$$
\frac{\partial^2 y_m^i}{\partial B_m^i \partial p_m^i} = \frac{\partial^2 y_m^i}{\partial p_m^i \partial B_m^i} = -\frac{S_m^i}{\left(B_m^i\right)^2} \left(1 - \Gamma_m^i\right). \tag{23}
$$

Then, for Hessian matrix H, the leading principal minor of order one is given by

$$
\Delta_1 = \frac{\partial^2 y_m^i}{\partial \left(B_m^i\right)^2} = \frac{2 S_m^i p_m^i}{\left(B_m^i\right)^3}. \tag{24}
$$

It clear that $\Delta_1 > 0$.

The leading principal minor of order two is

$$
\begin{aligned}
\Delta_2 &= \frac{\partial^2 y_m^i}{\partial \left(B_m^i\right)^2} \frac{\partial^2 y_m^i}{\partial \left(p_m^i\right)^2} - \frac{\partial^2 y_m^i}{\partial B_m^i \partial p_m^i} \frac{\partial^2 y_m^i}{\partial p_m^i \partial B_m^i} \\
&= \frac{(S_m^i)^2}{\left(B_m^i\right)^4} \left(\Omega_m^i - \left(1 - \Gamma_m^i\right)^2 \right),
\end{aligned} \tag{25}
$$

where

$$\Omega_m^i = 2\Gamma_m^i\left(2\Gamma_m^i\left(1 + x_m^i\right) - \frac{1}{\left(1 + x_m^i\right)} - 1\right). \tag{26}$$

Derived from (26), $\partial\Omega_m^i/\partial x_m^i > 0$, which implies that Ω_m^i is monotonically increasing with x_m, and Ω_m^i has a minimum value when $x_m^i = 1$. We have

$$\Omega_m^{\min} = \frac{1}{\ln 2}\left(\frac{4 - 3\ln 2}{2\ln 2}\right) > 1$$

and $0 < (1 - \Gamma_m)^2 < 1$. Hence, $\Delta_2 > 0$.

Hessian matrix H is positive definite since the all leading principal minors of matrix H are larger than 0. The objective function y_m^i is convex. The constraint qualifications of (19) are convex. As a result, the problem in (19) is a convex optimisation problem. □

Owing to (19) is convex. Similarly to (13), the partial Lagrange function of (19) is formulated in (27), where $\mu \geq 0$ and $\boldsymbol{\eta} \geq \mathbf{0}$ are the Lagrange multipliers related to $C3$ and $C5$ respectively.

$$\begin{aligned} &L(\boldsymbol{B}, \boldsymbol{P}, \mu, \boldsymbol{\eta}) \\ &= \sum_{i \in \mathscr{J}} \sum_{m \in \mathscr{M}_i} e_m p_m^i \frac{D_m}{B_m^i \log_2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)} \\ &+\mu\left(\sum_{m \in \mathscr{M}_i} B_m^i - B^i\right) + \eta_m\left(\frac{C_m}{f_m^i} + \frac{D_m}{r_m^i} - t_m^{\max}\right). \end{aligned} \tag{27}$$

Taking the partial derivative of $L(\boldsymbol{B}, \boldsymbol{P}, \mu, \boldsymbol{\eta})$ with respect to B_m^{i*}, $m \in \mathscr{M}_i$, and sets it equal 0. Subsequently, the fraction of the bandwidth can be formulated as

$$B_m^{i*} = \sqrt{\frac{(e_m p_m + \eta_m)D_m}{\mu \log_2\left(1 + x_m^i\right)}}. \tag{28}$$

The partial derivative of $L(\boldsymbol{B}, \boldsymbol{P}, \mu, \boldsymbol{\eta})$ with respect to μ is

$$\frac{\partial L}{\partial \mu} = \sum_{m \in \mathscr{M}_i} B_m^i - B^i = 0. \tag{29}$$

Based on (28) and (29), it can be known that

$$\mu = \left(\frac{1}{B^i} \sum_{m \in \mathscr{M}_i} \sqrt{\frac{\left(e_m p_m^i + \eta_m\right)D_m}{\log_2\left(1 + x_m^i\right)}}\right)^2. \tag{30}$$

Then, by inserting (30) into (28), the optimal bandwidth allocation B_m^{i*} for MD m is obtained, which is given by

$$B_m^{i*} = \frac{B^i}{\sum_{m \in \mathscr{M}_i} \sqrt{S_m^i p_m^i + \frac{\eta_m D_m}{\log_2\left(1 + x_m^i\right)}}} \sqrt{S_m^i p_m^i + \frac{\eta_m D_m}{\log_2\left(1 + x_m^i\right)}}. \tag{31}$$

In a similar way, we calculate the partial derivative of $L(\boldsymbol{B}, \boldsymbol{P}, \mu, \boldsymbol{\eta})$ with respect to p_m^i, and it is shown as below

$$\begin{aligned} \frac{\partial L}{\partial p_m^{i*}} &= \frac{e_m D_m}{B_m^i \log_2(1 + \frac{p_m^i g_m^i}{\delta_i^2})}\left(1 - \frac{p_m^i g_m^i/\delta_i^2}{(1 + \frac{p_m^i g_m^i}{\delta_i^2})\log_2(1 + \frac{p_m^i g_m^i}{\delta_i^2})\ln 2}\right) \\ &-\eta_m \frac{e_m D_m g_m^i/\delta_i^2}{B_m^i\left(\log_2(1 + \frac{p_m^i g_m^i}{\delta_i^2})\right)^2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)\ln 2} = 0.. \end{aligned} \tag{32}$$

```
Initialization:
    Computation task $\varphi_m = \{C_m, D_m, t_m^{max}\}$, $m \in \mathcal{M}_i$, maximum tolerance precision $\varepsilon_2 > 0$, $\eta_m^{\min}$, $\eta_m^{\max}$.
while $\eta_m^{\max} - \eta_m^{\min} > \varepsilon_2$ do
    Define $\eta_m = \left(\eta_m^{\min} + \eta_m^{\max}\right)/2$, compute $p_m^i$ and $B_m^i$ according to substitute $\eta_m$ into (32) and (31) respectively.
    If $t_m^i < t_m^{max}$, update $\eta_m^{\max} = \eta_m$, otherwise update $\eta_m^{\min} = \eta_m$.
end while
Output: The optimal transmission power $\mathbf{P}^*$ and the optimal bandwidth allocation $\mathbf{B}^*$.
```

Fig. 3 *Algorithm 2: Optimal Algorithm for Solving Problem (19)*

```
Initialization:
    Computation task $\varphi_m = \{C_m, D_m, t_m^{max}\}$, $m \in \mathcal{M}_t$, maximum tolerance precision $\varepsilon_3 > 0$, $l = 0$, $p_m(0) = p_m^{max}$, $B_m^i(0) = \frac{B^i}{M_i}$, based on Algorithm 1, obtain computation resource allocation $f_m(0)$, calculate $\Psi\left(\mathbf{f}(0), \mathbf{B}(0), \mathbf{P}(0)\right)$.
while $|\Theta(l) - \Theta(l-1)| > \varepsilon_3$ do
    Based on Algorithm 1, obtain the computation resource allocation $f_m^i(l)$. Based on Algorithm 2, obtain transmission power $p_m^i(l)$ and bandwidth allocation $B_m^i(l)$.
    Calculate $\Theta\left(\mathbf{f}(l), \mathbf{B}(l), \mathbf{P}(l)\right)$.
    $l = l + 1$
end while
Output: The optimal computation resource allocation $\mathbf{f}^*$, transmission power $\mathbf{P}^*$, and bandwidth allocation $\mathbf{B}^*$.
```

Fig. 4 *Algorithm 3: Jointly optimise computation resource, bandwidth and transmission power allocation for solve problem (12)*

The optimal transmission power can be obtained by solving Lagrange multiplier η_m in (32). Specially, we need to guarantee that the transmission power satisfies constraint $C3$. Based on (32), we can achieve that

$$\eta_m = \frac{e_m \delta_i^2}{g_m^i}\left(\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)\log_2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)\ln 2 - \frac{p_m^i g_m^i}{\delta_i^2}\right). \tag{33}$$

The partial derivation of η_m with respect to p_m^i is shown as

$$\frac{\partial \eta_m}{\partial p_m^i} = e_m \log_2\left(1 + \frac{p_m^i g_m^i}{\delta_i^2}\right)\ln 2. \tag{34}$$

From (34), we can know that $\partial\eta_m/\partial p_m^i > 0$. Since η_m is monotonically increasing with p_m^i, we can easily obtain the range of η_m according to constraint $C3$, where $\eta_{\mathrm{m}}^{\min} = \eta_m(p_m^{\min})$ and $\eta_{\mathrm{m}}^{\max} = \eta_m(p_m^{\max})$. Then, the optimal bandwidth allocation B_m^{i*} and transmission power p_m^{i*} are obtained by applying the bisection algorithm, which is detailed in Algorithm 2 (see Fig. 3).

Combining computation resource allocation, bandwidth allocation and transmission power assignment, Algorithm 3 (see Fig. 4) is proposed to solve problem (12) for a given computation offloading strategy. Initially, base on the given computation offloading strategy, the bandwidth of WAPs is fairly allocated to offloading MDs. Next, based on the initial bandwidth and transmission power, the initial computation resource allocation is obtained. Then, Algorithm 3 (Fig. 4) goes into iterate process, where computing resource allocation and radio resource (bandwidth and transmission power) assignment are iteratively updated based on Algorithms 1 and 2 (Figs. 2 and 3), respectively. When the iterating process is terminated, the optimal resource allocation is achieved.

3.3 Joint optimisation of computation offloading decision making, computation resource allocation, bandwidth allocation, and transmission power assignment

In this sub-section, the CSAO scheme is detailed to solve problem (8), where the optimal offloading decision making, computation

```
Initialization:
  Computation task: $\varphi_m = \{C_m, D_m, t_m^{max}\}$, $m \in \mathcal{M}$, initial offloading decision $a_m = 0$,($m \in \mathcal{M}$), each MD has opportunity to update the offloading decision $update_m^t = 1$
  Repeat:
for all MD $m \in \mathcal{M}$ and $update_m^t = 1$ do
  for all wireless AP $i \in \mathcal{J}$ do
    Based on Algorithm 3, allocate computation resource, transmission power and bandwidth for $\left(a'_m, a_{-m}(t)\right)$, and calculate $z_m^i\left(a'_m, a_{-m}(t)\right)$
    if $z_m\left(a'_m, a_{-m}(t)\right) < z_m\left(a_m, a_{-m}(t)\right)$ then
      $i \to \widetilde{X}_m(t)$
    else
      $z_m\left(a'_m, a_{-m}(t)\right) = \infty$
    end if
  end for
  Obtain the best offloading update set $X_m(t)$
end for
if $X_m(t) = \emptyset$ then
  $update_m^0 = 0$, $a_m^t = a_m^{t-1}$
else
  MD contends the update opportunity based on the offloading update set $\mathbf{X}(\mathrm{t}) = \{X_1(t), ..., X_M(t)\}$
  If cost of MD $m$ is minimal, update offloading strategy of MD $m$, $a_m = i_m^*$, and $m \to \mathcal{M}_i$, $\mathcal{M} = \mathcal{M} \backslash m$, $update_m^t = 0$, otherwise $a_m^t = a_m^{t-1}$.
end if
Until: Convergence
```

Fig. 5 *Algorithm 4: The Algorithm for CSAO*

resource allocation, transmission power assignment, and bandwidth allocation are obtained for minimising system cost. Concretely, at first, every MD is allowed to update its computation offloading decision ($\text{update}_m^t = 1$). MD m first computes its cost z_m^i based on offloading decision $(a'_m, a_{-m}(t))$, and it obtains its offloading update set $\widetilde{X}_m$. Then, based on $\widetilde{X}_m$, MD m computes its set of best offloading update as $X_m(t) \triangleq \{i = \arg\min z_m^i(a'_m, a_{-m}(t))$ and $z_m(a'_m, a_{-m}(t)) < z_m(a_m, a_{-m}(t))\}$. After that, each MD that satisfies $X_m(t) \neq \emptyset$ contends for the update opportunity. Moreover, the MD with greatest reduction in system costs wins the competition. This MD updates its computation offloading decision, and it is removed from set $\mathscr{M}$ into set $\mathscr{M}_i$. The whole process is repeated until convergence. According to the above process, the optimal solutions are obtained, where we achieve NE of computation offloading game and no MD deviates from this offloading strategy. For sake of a clear understanding, the whole procedure of CSAO scheme is detailed in Algorithm 4 (see Fig. 5).

Then, the computational complexity of the CSAO scheme is analysed in the following. The computational complexity of the CSAO scheme in Algorithm 4 (Fig. 5) mainly come from Algorithm 3 (Fig. 4). Algorithm 3 (Fig. 4) consists of Algorithms 1 and 2 (Fig. 2 and 3). In Algorithm 1 (Fig. 2), the subgradient projection method needs $O(1/\varepsilon_1^2)$ iterations to converge. The complexity of Algorithm 2 (Fig. 3) is $O\left(\frac{\eta^{\max} - \eta^{\min}}{\varepsilon_2}\right)$. The while-loop in Algorithm 3 (Fig. 4) needs $O(1/\varepsilon_3^2)$ iterations to converge, thus the computational complexity of Algorithm 3 (Fig. 4) is

$$O\left(\frac{1}{\varepsilon_3^2}\left(\frac{1}{\varepsilon_1^2} + \frac{\eta^{\max} - \eta^{\min}}{\varepsilon_2}\right)\right).$$

In addition, at the each iteration in Algorithm 4 (Fig. 5), the potential game needs $O(T)$ calculations to obtain the offloading update set of MDs. Therefore, the computation complexity of CSAO is

$$O\left(KT\frac{1}{\varepsilon_3^2}\left(\frac{1}{\varepsilon_1^2} + \frac{\eta^{\max} - \eta^{\min}}{\varepsilon_2}\right)\right),$$

where K denotes the number of iterations of Algorithm 4 (Fig. 5).

In Section 3, we decompose the formulated problem (8) into two sub-problems: (i) the computation offloading decision problem with involving the WAP selection, and (ii) the resource allocation problem including the computation resource allocation, the transmission power assignment, and the bandwidth allocation. By solving these subproblems sequentially in each iteration until convergence, the optimal solution for problem (8) can be achieved with a low computation complexity.

4 Numerical results and discussion

In this section, representative numerical results are presented to evaluate the performance of the proposed scheme. Initially, the convergence performances of the four algorithms (Algorithms 1, 2, 3 and 4 (Figs. 2–5)) are validated in Section 4.1. Then, the performance of our proposed CSAO scheme is evaluated in Section 4.2.

In the numerical analysis, a multiple WAPs network with MEC is considered, where we assume that all MDs located in a building with 10 WAPs [26]. For each MD, the maximum transmission power and the maintaining power are respectively set as 0.4 and 0.05 W [26]. The impact factor of energy consumption and monetary cost are set as 2 and 0.01, respectively. Moreover, the data size of the tasks and computing requirement obey the Guass distribution with mean $\mu_d = 350$ KB and $\mu_c = 3$ Ghz, and standard deviation $\delta_d = 84$ KB and $\delta_c = 0.6$ Ghz, respectively. The total computing resource of MEC server and the unit cost of computation resource are 2000 Ghz and 0.1 $Ghz/\$$, respectively. The local computation capacity of MDs follows a normal distribution with mean $\mu_l = 1$ Ghz and standard deviation $\delta_l = 0.5 * \mu_l$.

4.1 Convergence of algorithms 1, 2, 3, and 4

In this sub-section, we present the convergence performances of the proposed algorithms, such as Algorithms 1, 2, 3, and 4 (Figs. 2–5). First, we evaluate the convergence evolution of the inner loop of Algorithm 3 (Fig. 4), and it includes Algorithms 1 and 2 (Fig. 2 and 3). Then, we demonstrate the convergence of Algorithms 3 and 4 (Fig. 4 and 5).

Fig. 6*a* plots system cost versus the number of iterations under different computing resource of the MEC server, which depicts the convergence evolution of Algorithm 1 (Fig. 2). Algorithm 1 (Fig. 2) adjusts λ and $\boldsymbol{\beta}$ to achieve the optimal computation resource allocation while meeting the constraint of computing resource of MEC and the maximum tolerable delay of the computation task, which seeks to minimise system cost under given wireless resource. Fig. 6*b* presents the convergence of Algorithm 2 (Fig. 3) for different available bandwidth, where the simulations are performed at WAP 5 with 25 MDs. That is, 25 MDs offload their computation tasks to the MEC server via WAP 5, and radio resource allocation in Algorithm 2 (Fig. 3) is performed among 25 MDs. Moreover, it should be noted that Algorithm 2 (Fig. 3) needs a few iterations to converge.

Figs. 7*a* and 7*b* show the convergence of the outer loop of Algorithm 3(Fig. 4), where there are 50 and 25 MDs in Fig. 7*a* and Fig. 7*b*, respectively. Moreover, it is worth noting that Algorithm 3 (Fig. 4) has a fast convergence rate, which converges typically in several iterations. Fig. 7*c* highlights the convergence performance of Algorithm 4 (Fig. 5) under different number of MDs. As shown in Fig. 7*c*, it is observed that the proposed scheme can keep system cost decreasing after each iteration until convergence to a low and stable state, which is a NE. This is because the proposed CSAO scheme seeks to minimise system cost in each iteration by optimising computation offloading decision making, transmission power, bandwidth and computation resource allocation, thus system cost can be decreased.

Fig. 8 depicts the number of iterations for the convergence of our proposed CSAO scheme with different the number of MDs, which evaluates the scalability of Algorithm 4 (Fig. 5). As depicted in Fig. 8, the number of iterations for the convergence of Algorithm 4 (Fig. 5) shows a weak-linear growth trend as the

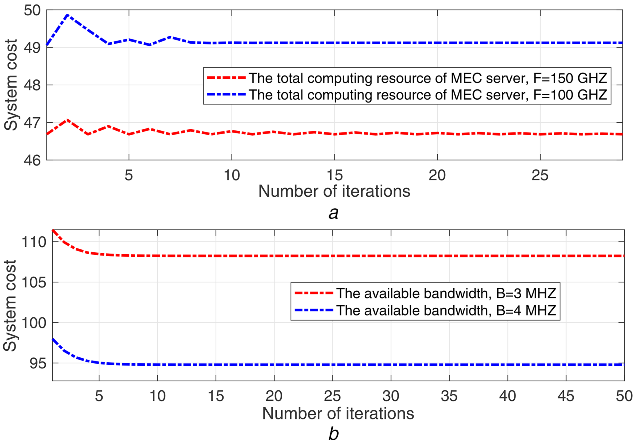

Fig. 6 *Convergence of Algorithms 1 and 2*
(a) Convergence of Algorithm 1 (Fig. 2), ***(b)*** Convergence of Algorithm 2 (Fig. 3)

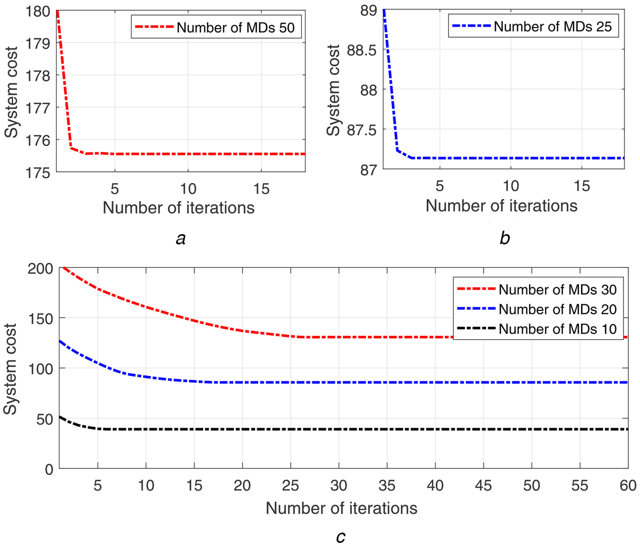

Fig. 7 *Convergence of Algorithms 3 and 4*
(a) Convergence of Algorithm 3 (Fig. 4) under 50 MDs, ***(b)*** Convergence of Algorithm 3 (Fig. 4) under 25 MDs, ***(c)*** Convergence of Algorithm 4 (Fig. 5)

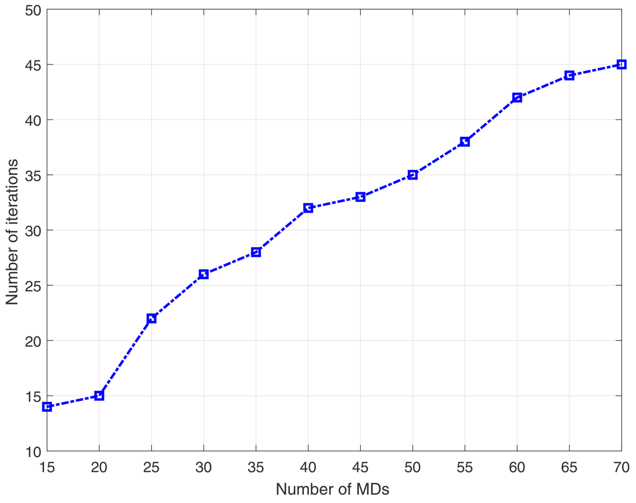

Fig. 8 *Number of iterations for the convergence of CSAO scheme with different number of MDs*

number of MDs increases. This result demonstrates that the convergence of Algorithm 4 (Fig. 5) is fast and stable, and it scales well with the increasing of the number of MDs.

4.2 Performance of CSAO scheme

Next, we evaluate the performance of the proposed CSAO scheme in comparison with the following three schemes.

- *CDO* scheme [26]. The computation offloading decision optimisation (CDO) scheme only selects the optimal WAP to offload computation task as described in Section 3.1 for minimising system cost, where computing and wireless resource are fairly allocated to all MEC processing MDs.
- *RAO* scheme. The resource allocation optimisation (RAO) scheme jointly optimises the computation resource allocation, the transmission power, and the bandwidth allocation as described in Section 3.2, where each MEC processing MD randomly selects one of the WAPs to offload its computation task.
- *ECO* scheme. The enumeration computation offloading optimisation (ECO) scheme explores all offloading decision combinations, and chooses an optimal offloading decision combination to minimise system cost for each MD, which is able to obtain an optimal solution for problem (8) .

Fig. 9 plots the comparison of system cost with respect to the number of MDs under all schemes (CSAO, CDO, RAO and ECO schemes). As shown in Fig. 9, although system cost increases with the growth of the number of MDs for all schemes, the performance of the proposed CSAO scheme significantly outperforms CDO and RAO schemes. The reason is that the CDO scheme only optimises WAP selection, and the RAO scheme only optimises the computing resource, transmission power and bandwidth allocation. For the CDO scheme, the computation and wireless resource is evenly allocated to all MEC processing MDs. While each MD requires different computing resources for processing computation task, and each computation task has different input data and delay constraints. Hence, the CDO scheme can lead to an increase in the system cost. For the RAO scheme, every MD randomly selects a WAP to migrate its computation task, resulting in an unbalance load among WAPs and a high system cost. Especially, the unbalance load among WAPs can become more obvious as the number of MDs increases. In that case, therefore, the system cost gap becomes larger between our proposed CSAO scheme and RAO scheme. The proposed CSAO scheme balances the amount of offloading MDs on each WAP, and jointly optimises offloading decision making, computation resource allocation, transmission power and bandwidth assignment. In addition, despite the system cost of the ECO scheme is slightly lower than the proposed CSAO scheme, the ECO scheme has very high algorithm complexity. To be specific, the ECO scheme traverses all offloading decision combinations, and each MD has $I+1$ offloading sections. Consequently, the computation complexity of ECO scheme is

$$O\left((I+1)^M \frac{1}{\varepsilon_3^2}\left(\frac{1}{\varepsilon^2}+\frac{\eta^{\max}-\eta^{\min}}{\varepsilon_1}\right)\right).$$

The computation complexity of the proposed CSAO scheme is

$$O\left(KT\frac{1}{\varepsilon_3^2}\left(\frac{1}{\varepsilon_1^2}+\frac{\eta^{\max}-\eta^{\min}}{\varepsilon_2}\right)\right),$$

which is analysed in Section 3.3. The proposed scheme can achieve a near-optimal solution and has a higher computational efficiency compared to the ECO scheme.

The impact of required computing resource of computation task on system cost of three schemes (CSAO, CDO and RAO schemes) is shown in Fig. 10, where the simulations are performed under 30 MDs. From Fig. 10, it can be observed that three schemes have an approximate system cost when the required computing resource is lower than 1.6 Ghz. The main reason is that the MEC processing results in more cost than local processing when the required computing resource of the computation task is low. In this case, the computation resource of MDs can fully meet the demand of computation. After that, as required computing resource of computation task increases, more MDs intend to utilise the

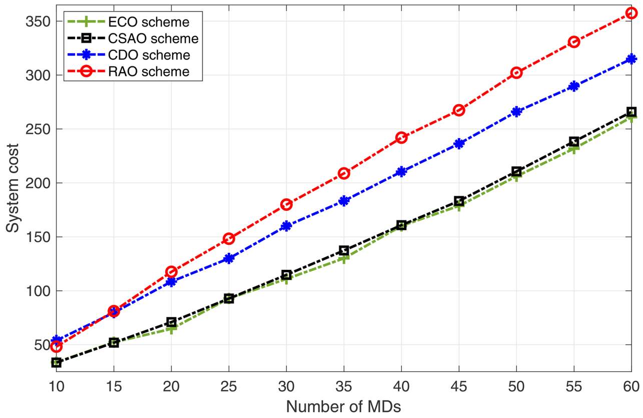

Fig. 9 *System cost versus the number of MDs*

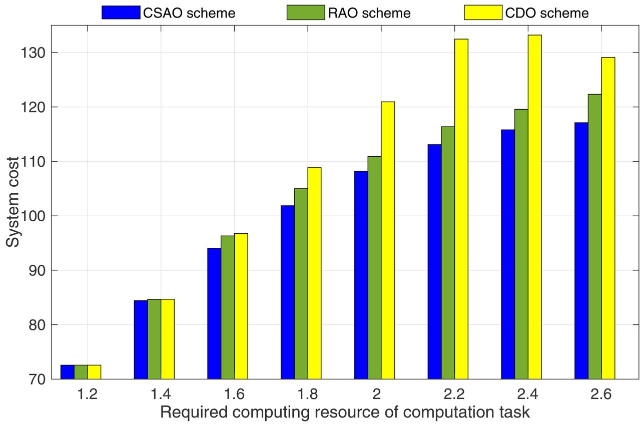

Fig. 10 *System cost versus required computing resource of computation task*

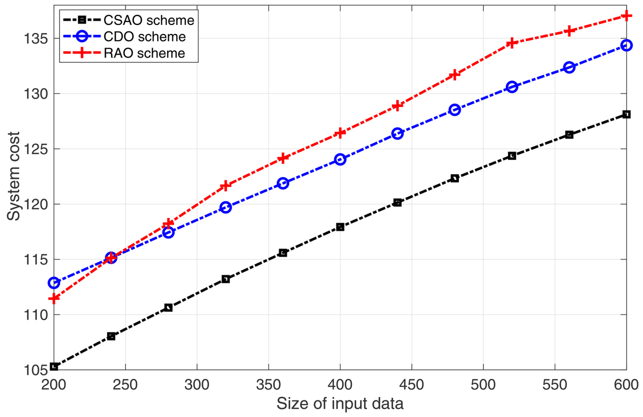

Fig. 11 *System cost versus size of input data*

computation resource of MEC via computation offloading to mitigate the heavy cost of local computing. When many MDs offload their computation tasks to MEC, the computation resource of the MEC server, and radio resources between MDs and WAPs are shared, which should be allocated precisely to reduce system cost. Therefore, it can be noted that the system cost of CDO scheme mounts up rapidly while the RAO and the proposed CSAO schemes are relatively slow in Fig. 10, where the proposed CSAO and RAO schemes properly allocate computing and communication resources based on different computation tasks, while the CDO scheme does not consider resource allocation. Additionally, compared to RAO and CDO schemes, the proposed CSAO scheme is able to save more system cost with required computing resource of computation task increasing. Since the proposed CSAO scheme can make full use of the wireless resource of all WAPs, and optimise the computation offloading decision and resource allocation to further decrease system cost.

Fig. 11 shows the comparative results with the different system cost among CSAO, CDO and RAO schemes, where we implement the simulations with 30 MDs. As shown in Fig. 11, it can be seen that system cost keeps adding with the increase of data size for all schemes. When the data size of computation task is relatively small, the system cost of the RAO scheme is less than the CAO scheme. After that, with the data size increasing, the CDO scheme has a lower system cost compared to the RAO scheme. While our proposed CSAO scheme always achieves the lowest system cost in comparison with CDO and RAO schemes. Moreover, from the figure, we can know that the difference in system cost between RAO scheme and other schemes becomes more obvious when the data size of the computation task becomes larger. This is because the load balancing among WAPs can be achieved in the proposed CSAO and CDO schemes. According to load balancing, many MDs avoid selecting the same WAP to transmit input data. In other words, CDO and CSAO schemes make the best use of all WAPs for improving the data transmission rate in the computation offloading process. Therefore, compared to the RAO schemes, the CDO and the proposed CSAO schemes significantly degrade system cost as the input data size increases. In addition, although the CDO scheme balances the load of WAPs, the proposed CSAO schemes can obtain a lower system cost in comparison with CDO scheme. Since, the proposed scheme not only balances the load of WAPs, but also jointly optimises computing and wireless resources allocation.

In summary, we formulate the WAP selection as an offloading decision in this paper, and the load balance among WAPs can be realised by optimising the computation offloading strategy. Moreover, based on the simulation results, we can obviously observe that the joint optimisation of offloading decision making, computation resource allocation, bandwidth allocation and transmit power assignment is indispensable and important in a multiple WAPs network with MEC. The proposed scheme is efficient and feasible in dropping the system cost, especially when the number of MDs increases or the size of input data is higher.

5 Conclusions

This paper is interested in solving the computation offloading problem in a multiple WAPs network with MEC, where every MD can either process its computation locally or select one WAP among multiple WAPs to offload its computation task to MEC. Specially, we model WAP selection as an offloading decision to analyse the load balancing among WAPs. Then, this problem is formulated as a constrained optimisation problem for minimising system cost. By joint optimisation computation offloading and WAP selection, a CSAO scheme is developed for obtaining the optimal solution. Further, computation resource allocation, transmission power assignment, and bandwidth allocation are also taken into consideration in this scheme. Through our simulation results, it is clear that the proposed CSAO scheme substantially decreases the computation complexity without loss of the performance. Moreover, the proposed CSAO scheme achieves the lower system cost in comparison with CAO and RAO schemes. The advantage is more obvious when there are many resource hungry computation-intensive tasks. In addition, it is also obvious that the proposed CSAO scheme converges in a fast manner and scales well as the MD size increases.

In this paper, we consider that all MDs remain stationary during an offloading period. While MDs may depart and leave dynamically within a computation offloading period. Therefore, for future work, we are going to design the more general computation offloading schemes, where the MD mobility patterns will be considered. Besides, this paper considers the case that all WAPs are linked to one MEC server. With the advancements in computing hardware, MEC servers may be densely distributed. Offloading policies can be further developed for a multiple WAPs network with multiple MEC servers in the future.

6 Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2018YJS008, the National

Natural Science Foundation of China under Grant 61661021, National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2016ZX03001014-006, Beijing Natural Science Foundation under Grant L182018, the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University under Grant 2017D14, Science and Technology Project of Jiangxi Provincial Transport Bureau under Grant 2016D0037, Science and Technology Program of Jiangxi Province under Grants 20172BCB22016 and 20171BBE50057, Shenzhen Science and Technology Program under Grant JCYJ20170817110410346.

7 References

[1] Mao, Y., Zhang, J.B., Letaief, K.: 'Dynamic computation offloading for mobile-edge computing with energy harvesting devices', *IEEE J. Sel. Areas Commun.*, 2016, **34**, (12), pp. 3590–3605

[2] Zhao, J., Liu, Y., Gong, Y., *et al.*: 'A dual-link soft handover scheme for c/u plane split network in high-speed railway', *IEEE Access*, 2018, **6**, pp. 12473–12482

[3] Abbas, N., Zhang, Y., Taherkordi, A., *et al.*: 'Mobile edge computing: a survey', *IEEE Internet Things J.*, 2018, **5**, (1), pp. 450–465

[4] Zhang, J., Xia, W., Zhang, Y., *et al.*: 'Joint offloading and resource allocation optimization for mobile edge computing'. Proc. IEEE Global Commun. Conf. (GLOBECOM), Singapore, December 2017, pp. 1–6

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

[6] Lu, W., Gong, Y., Liu, X., *et al.*: 'Collaborative energy and information transfer in green wireless sensor networks for smart cities', *IEEE Trans. Ind. Inf.*, 2018, **14**, (4), pp. 1585–1593

[7] Asghari, S., Navimipour, N.J.: 'Nature inspired meta-heuristic algorithms for solving the service composition problem in the cloud environments', *Int. J. Commun. systs.*, 2018, **31**, (12), p. e3708

[8] Baker, T., Al-Dawsari, B., Tawfik, H., *et al.*: 'GreeDi: an energy efficient routing algorithm for big data on cloud', *Ad Hoc Netw.*, 2015, **35**, pp. 83–96

[9] Naseri, A., Navimipour, N.J.: 'A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm', *J. Ambient Intell. Humaniz. Comput.*, 2019, **10**, (5), pp. 1851–1864

[10] Baker, T., Ngoko, Y., Tolosana-Calasanz, R., *et al.*: 'Energy efficient cloud computing environment via autonomic meta-director framework'. Proc. IEEE Int. Conf. Developments in eSystems Engineering (DeSE), Abu Dhabi, UAE, December 2013, pp. 198–203

[11] Sheikholeslami, F., Navimipour, N.J.: 'Auction-based resource allocation mechanisms in the cloud environments: A review of the literature and reflection on future challenges', *Concurrency Computat. Pract. Exper.*, 2018, **30**, (16), p. e4456

[12] Vakili, A., Navimipour, N.J.: 'Comprehensive and systematic review of the service composition mechanisms in the cloud environments', *J. Netw. Comput. Appl.*, 2017, **81**, pp. 24–36

[13] Azad, P., Navimipour, N.J.: 'An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm', *Int. J. Cloud Appl. Comput. (IJCAC)*, 2017, **7**, (4), pp. 20–40

[14] Al-khafajiy, M., Baker, T., Waraich, A., *et al.*: 'Iot-fog optimal workload via fog offloading'. Proc. IEEE/ACM Int. Conf. on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, December 2018, pp. 359–364

[15] Li, Q., Zhao, J., Gong, Y., *et al.*: 'Energy-efficient computation offloading and resource allocation in fog computing for internet of everything', *China Commun.*, 2019, **16**, (3), pp. 32–41

[16] Al-khafajiy, M., Baker, T., Al-Libawy, H., *et al.*: 'Fog computing framework for internet of things applications'. Proc. IEEE Proc. IEEE Int. Conf. Developments in eSystems Engineering (DeSE), Cambridge, UK, September 2018, pp. 71–77

[17] Al-khafajiy, M., Baker, T., Al-Libawy, H., *et al.*: 'Improving fog computing performance via fog-2-fog collaboration', *Future Gener. Comput. Syst.*, 2019, **100**, pp. 266–280

[18] Tao, X., Ota, K., Dong, M., *et al.*: 'Performance guaranteed computation offloading for mobile-edge cloud computing', *IEEE Wirel. Commun. Lett.*, 2017, **6**, (6), pp. 774–777

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

[20] 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

[21] Feng, J., Zhao, L., Du, J., *et al.*: 'Computation offloading and resource allocation in D2D-enabled mobile edge computing'. Proc. IEEE Int. Conf. Commun. (ICC), Kansas City, MO, July 2018, pp. 1–6

[22] Ma, X., Zhang, S., Li, W., *et al.*: 'Cost-efficient workload scheduling in cloud assisted mobile edge computing'. Proc. IEEE/ACM 25th Int. Symp. Quality of Service (IWQoS), Vilanova i la Geltru, Spain, June 2017, pp. 1–10

[23] Chen, M.-H., Liang, B., Dong, M.: 'Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point'. Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), Atlanta, GA, USA, May 2017, pp. 1–9

[24] Du, J., Zhao, L., Feng, J., *et al.*: 'Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee', *IEEE Trans. Commun.*, 2018, **66**, (4), pp. 1594–1608

[25] Vu, T.T., Van Huynh, N., Hoang, D.T., *et al.*: 'Offloading energy efficiency with delay constraint for cooperative mobile edge computing networks'. Proc. IEEE Global Commun. Conf. (GLOBECOM), Abu Dhabi, United Arab Emirates, December 2018, pp. 1–6

[26] Ma, X., Lin, C., Xiang, X., *et al.*: 'Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing'. Proc. ACM Int. Conf. Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Cancun, Mexico, November 2015, pp. 271–278

[27] Zhang, J., Xia, W., Cheng, Z., *et al.*: 'An evolutionary game for joint wireless and cloud resource allocation in mobile edge computing'. Proc. 9th Int. Conf. Wireless Commun. Signal Processing (WSCP), Nanjing, China, October 2017, pp. 1–6

[28] Wang, Y., Sheng, M., Wang, X., *et al.*: 'Mobile-edge computing: partial computation offloading using dynamic voltage scaling', *IEEE Trans. Commun.*, 2016, **64**, (10), pp. 4268–4282

[29] Zhang, K., Mao, Y., Leng, S., *et al.*: 'Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks', *IEEE Access*, 2016, **4**, pp. 5896–5907

[30] Zhao, J., Yang, T., Gong, Y., *et al.*: 'Power control algorithm of cognitive radio based on non-cooperative game theory', *China Commun.*, 2013, **10**, (11), pp. 143–154