Joint Task Offloading, CNN Layer Scheduling and Resource Allocation in Cooperative Computing System

Xia Song(✉), Rong Chai, and Qianbin Chen

Key Lab of Mobile Communication Technology,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2360321633@qq.com, {chairong,chenqb}@cqupt.edu.cn

Abstract. In this paper, we consider a cooperative computing system which consists of a number of mobile edge computing (MEC) servers deployed with convolutional neural network (CNN) model, a remote mobile cloud computing (MCC) server deployed with CNN model and a number of mobile devices (MDs). We assume that each MD has a computation task and is allowed to offload its task to one MEC server where the CNN model with various layers is applied to conduct task execution, and one MEC server can accept multiple tasks of MDs. To enable the cooperative between the MEC servers and the MCC server, we assume that the task of MD which has been processed partially by the CNN model of the MEC server will be sent to CNN model of the MCC server for further processing. We study the joint task offloading, CNN layer scheduling and resource allocation problem. By stressing the importance of task execution latency, the joint optimization problem is formulated as an overall task latency minimization problem. As the original optimization problem is NP hard, which cannot be solved conveniently, we transform it into three subproblems, i.e., CNN layer scheduling subproblem, task offloading subproblem and resource allocation subproblem, and solve the three subproblems by means of extensive search algorithm, reformulation-linearization-technique (RLT) and Lagrangian dual method, respectively. Numerical results demonstrate the effectiveness of the proposed algorithm.

Keywords: Cooperative computing · MEC server · MCC server · CNN layer scheduling · Task offloading

1 Introduction

The rapid development of mobile Internet and smart devices promotes the emergence of new applications such as interactive gaming, virtual reality, augment reality, etc. However, the intensive computing requirements of these emerging applications pose great challenges to the computation and process capability of

H. Gao et al. (Eds.): ChinaCom 2019, LNICST 312, pp. 129–142, 2020.
https://doi.org/10.1007/978-3-030-41114-5_11

mobile devices (MDs). While mobile cloud computing (MCC) can be applied to address these challenges, it suffers from long latency and low efficiency for transmitting and processing huge amounts of data collected from MDs [1]. To overcome the drawback of MCC, the concept of mobile edge computing (MEC) is proposed [2]. By deploying high performance MEC servers at the network edge in a distributed manner, MDs are allowed to offload their computation task to the MEC servers, which then execute the task on behalf of the MDs. Therefore, the task execution cost of the MDs especially in terms of task execution latency and energy consumption can be reduced significantly.

While acting as an efficient manner for task execution, the MEC servers may be subject to relatively limited computational capability especially compared to the remote MCC servers. Hence, cooperative computing system which enables the cooperative between remote MCC servers and MEC servers in task execution will be highly desired as it may enhance the performance of task execution and achieve the efficient resource utilization of the network. To further facilitate efficient task execution of the cooperative computing system, convolutional neural network (CNN) models can be applied at the MEC servers [3]. As a typical CNN model is composed of multiple tiers with each tier having various data processing capability, it is possible to process the task of MDs with a number of CNN layers, then transmit the reduced intermediate data to the MCC server to complete the task execution [4].

In recent years, the problem of task offloading has received considerable attentions [5–10]. To minimize the system-wide computation overheads, the authors in [5] formulate the task offloading problem as an offloading game and demonstrate the existence of Nash equilibrium point. Task offloading problem in MEC system was considered in [7,8], the weighted sum of the energy consumption and task execution delay was formulated and optimized in [7] and the maximum task execution latency of all the MDs was minimized in [8].

Joint task offloading and resource allocation problem was addressed in [6,9]. The authors in [6] defined an offloading priority function that depends on the local computing energy of the MDs and the channel gain between the MDs and the MEC servers. Based on the offloading priority of the MDs, a joint offloading decision and resource allocation strategy is designed to achieve the minimum weighted sum of the energy consumption. In [9], the problem of joint task offloading and resource allocation in an MEC system with multiple MDs was formulated as energy consumption minimization problem. To solve the formulated optimization problem, the authors further decoupled the original optimization problem into two problems, i.e., the resource allocation problem and the task offloading problem, and solved the two problems respectively by using the Lagrange method and the Hungarian method.

The aforementioned researches mainly address the problem of task offloading in MEC system, however, the cooperative between MCC schemes and MEC schemes has not been studied extensively. The authors in [10] considered the task offloading problem in a cooperative computing system serving one MD at certain time period, and proposed a greedy algorithm to maximize the number

of tasks which can be offloaded to the system successfully. The authors in [10] failed to consider the resource sharing at the computing server, thus may result in inefficient resource utilization and highly limited task offloading performance. Furthermore, the task execution time failed to be stressed, thus, may lead to relatively long task execution time, which is undesired, especially for delay-sensitive MD tasks.

In this paper, we consider a cooperative computing system which allows the cooperative between the MEC servers and the MCC server in task execution. Assuming that the task execution at each MEC server is conducted by a CNN model with multiple layer, we study the joint task offloading, CNN layer scheduling and resource allocation problem. The joint optimization problem is formulated as an overall task latency minimization problem. As the original optimization problem is NP hard, which cannot be solved conveniently, we transform it into three subproblems, i.e., CNN layer scheduling subproblem, task offloading subproblem and resource allocation subproblem, and solve the three subproblems by means of extensive search algorithm, reformulation-linearization-technique (RLT) and Lagrangian dual method, respectively.

2 System Model

In this paper, we consider a cooperative computing system which consists of N MEC servers, an MCC server and M MDs. Suppose each MEC server is deployed with a CNN model which is composed of one input layer, one output layer and a number of hidden layers. We assume that each MD has a single task which can be offloaded to one MEC server and the task execution at the MEC server is conducted by the CNN model. More specifically, the input data of MD task will be processed by the input layer and various hidden layers or/and output layer, and then is sent out at one particular hidden layer or the output layer of the CNN model. We further assume that the cooperative between the MEC servers and the MCC server in task execution is allowed, i.e., the MD task which has been processed partially by the CNN model will be sent to the MCC server for further processing. Figure 1 shows the system model considered in this paper.

Let MD_m denote the mth MD and $\mathrm{T} = \{\mathrm{T}_1, ..., \mathrm{T}_M\}$ denote the set of MD tasks, where T_m denotes the task of MD_m, $1 \leq m \leq M$. T_m can be characterized by a 3-tuple $\langle S_m, R_m^{\min}, D_m^{\max} \rangle$, where S_m denotes the size of input data of T_m, $R_m^{\min}$ and $D_m^{\max}$ denote the minimum transmission rate and the maximum tolerable task execution latency of T_m, respectively. We denote $\mathrm{E} = \{\mathrm{E}_1, ..., \mathrm{E}_N\}$ as the set of MEC servers, where E_n denotes the nth MEC server, $1 \leq n \leq N$. Let F_n denote the computation capability of E_n, B_n denote the bandwidth of the wireless link between E_n and the MDs, and C_n denote the capacity of the fronthaul link between E_n and the MCC server.

To improve the resource utilization of the MEC servers, we assume that multiple MDs are allowed to offload their task to one MEC server. However, in this case, the resource sharing between multiple MD tasks has to be considered. In particular, the bandwidth resource and computation resource of the MEC

servers and the capacity of the fronthaul link between the MEC servers and the MCC server should be allocated to various MD tasks.

We further denote r_n^k as the ratio of the intermediate data size generated at the kth CNN layer of E_n to the input data size of MD tasks and denote p_n^k as the computational overhead of a unit of input data at the kth CNN layer of E_n, $1 \leq n \leq N$, $1 \leq k \leq K$, where K denotes the total number of the layers in the CNN models employed at the MEC servers.

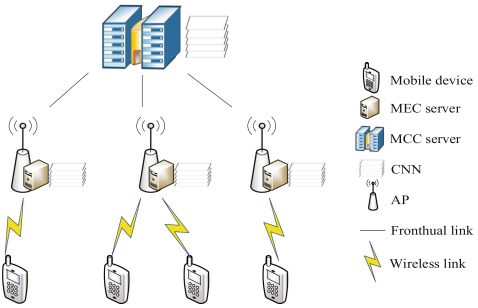

Fig. 1. System model

3 Optimization Problem Formulation

In this section, we examine the overall task latency in the considered cooperative computing system and formulate the joint task offloading, CNN layer scheduling and resource allocation problem as overall task latency minimization problem.

3.1 Objective Function

Stressing the performance of task execution in the cooperative computing system, we define overall task latency D as

$$D = \sum_{m=1}^{M} \sum_{n=1}^{N} x_{mn} D_{mn} \tag{1}$$

where D_{mn} denotes the task execution latency of T_m which is offloaded to E_n, x_{mn} denotes the task offloading variable of T_m. That is, if T_m is offloaded to E_n, $x_{mn} = 1$, otherwise, $x_{mn} = 0$. Jointly considering the time required to transmit MD tasks to the MEC servers and the MCC server, as well as the processing time at both servers, we formulate D_{mn} as

$$D_{mn} = D_{mn}^{\mathrm{t}} + D_{mn}^{\mathrm{p}} + D_{mn}^{\mathrm{tc}} + D_{mn}^{\mathrm{pc}} \tag{2}$$

where D_{mn}^{t} denotes the transmission latency required to offload T_m to E_n, D_{mn}^{p} denotes the task execution latency of T_m at E_n, D_{mn}^{tc} denotes the transmission

latency of T_m when being transmitted from E_n to the MCC server, D_{mn}^{pc} denotes the task executing latency of T_m at the MCC server.

D_{mn}^{t} in (2) can be formulated as

$$D_{mn}^{\mathrm{t}} = \frac{S_m}{R_{mn}} \tag{3}$$

where R_{mn} denotes the achievable data rate of T_m when being transmitted to E_n, which can be expressed as

$$R_{mn} = \alpha_{mn} r_{mn} \tag{4}$$

where $\alpha_{mn} \in [0, 1]$ denotes the fraction of the bandwidth resource allocated to T_m from E_n, r_{mn} denotes the data rate of T_m when being transmitted to E_n with the total bandwidth resource of E_n, i.e.,

$$r_{mn} = B_n \log_2 \left(1 + \frac{P_m g_{mn}}{\sigma^2}\right) \tag{5}$$

where P_m denotes the transmission power of MD_m when transmitting T_m to E_n, g_{mn} and σ^2 denote respectively the channel gain and the noise power of the link between MD_m and E_n.

D_{mn}^{p} in (2) can be calculated as

$$D_{mn}^{\mathrm{p}} = \sum_{k=1}^{K} \delta_{mn}^k p_n^k \frac{S_m}{\beta_{mn} F_n} \tag{6}$$

where δ_{mn}^k denotes CNN layer scheduling variable. If the first k layers of the CNN model is scheduled to process T_m at E_n, $\delta_{mn}^k = 1$, otherwise, $\delta_{mn}^k = 1$, $\beta_{mn} \in [0, 1]$ denotes the fraction of the computation capability allocated to T_m at E_n.

D_{mn}^{tc} in (2) can be calculated as

$$D_{mn}^{\mathrm{tc}} = \sum_{k=1}^{K} \delta_{mn}^k r_n^k \frac{S_m}{\lambda_{mn} C_n} \tag{7}$$

where $\lambda_{mn} \in [0, 1]$ denotes the fraction of the fronthaul capacity allocated to T_m when E_n transmits the intermediate data of T_m to the MCC server.

In this paper, we assume that the MCC server has relatively high computation capability, thus the latency term D_{mn}^{pc} in (2) is negligible.

3.2 Optimization Constraints

To design the optimal joint task offloading, CNN layer scheduling and resource allocation strategy which minimizes the overall task latency of the system, we should consider a number of constraints.

CNN Layer Scheduling Constraint. In this paper, we assume that at the MEC servers, one or multiple layer of the CNN model is assigned for executing one MD task, thus, we can express the CNN layer scheduling constraints as

$$\text{C1}: \ \delta_{mn}^{k} \in \{0,1\}, \tag{8}$$

$$\text{C2}: \sum_{k=1}^{K} \delta_{mn}^{k} \leq 1. \tag{9}$$

Task Offloading Constraint. We assume that each task can be offloaded to at most one MEC server, hence, the constraints can be

$$\text{C3}: \ x_{mn} \in \{0,1\}, \tag{10}$$

$$\text{C4}: \sum_{n=1}^{N} x_{mn} = 1, 1 \leq m \leq M. \tag{11}$$

Resource Allocation Constraints. In the case that one MEC server executes multiple MD tasks, the resource allocation constraints should be satisfied, which can be expressed as

$$\text{C5}: \ \alpha_{mn}, \ \beta_{mn}, \ \lambda_{mn} \in [0,1], \tag{12}$$

$$\text{C6}: \sum_{m=1}^{M} \alpha_{mn} \leq 1, \tag{13}$$

$$\text{C7}: \sum_{m=1}^{M} \beta_{mn} \leq 1, \tag{14}$$

$$\text{C8}: \sum_{m=1}^{M} \lambda_{mn} \leq 1. \tag{15}$$

Data Rate and Latency Requirements. Stressing the task offloading requirement, we assume that the link between MD_m and E_n should meet a minimum transmission rate constraint when T_m is offloaded to one MEC server, i.e.,

$$\text{C9}: \sum_{n=1}^{N} x_{mn} R_{mn} \geq R_{m}^{\min}, 1 \leq m \leq M. \tag{16}$$

As the task execution latency of T_m should meet a tolerable maximum latency requirement, we can express the task execution latency constraint as

$$\text{C10}: \sum_{n=1}^{N} x_{mn} D_{mn} \leq D_{m}^{\max}. \tag{17}$$

3.3 Optimization Problem

Considering the aforementioned objective function and optimization constraints, we formulate the overall task latency minimization-based joint task offloading, CNN layer scheduling and resource allocation problem as

$$\begin{aligned} \min_{x_{mn},\delta_{mn}^k,\alpha_{mn},\beta_{mn},\lambda_{mn}} \quad & D \\ \text{s.t.} \quad & \text{C1} - \text{C10}. \end{aligned} \tag{18}$$

Through solving above optimization problem, we can obtain the joint task offloading, CNN layer scheduling and resource allocation strategies.

4 Solution of the Optimization Problem

The formulated optimization problem is a non-convex mixed integer programming problem which is NP-hard and cannot be solved conveniently. In this section, we decompose the formulated optimization problem into three subproblems, i.e., CNN layer scheduling subproblem, task offloading subproblem and resource allocation subproblem, and solve the three sub-problems successively to obtain the joint task offloading, CNN layer scheduling and resource allocation strategies.

4.1 CNN Layer Scheduling Subproblem

In this subsection, we first assume that task offloading strategy is given, e.g., $x_{mn} = 1$, and no resource sharing among tasks is required, i.e., $\alpha_{mn} = 1$, $\beta_{mn} = 1$, $\lambda_{mn} = 1$. Substituting $\alpha_{mn},\ \beta_{mn},\ \lambda_{mn}$ into (2)–(7), we may rewrite D_{mn} as

$$D_{mn}^0 = \frac{S_m}{r_{mn}} + \sum_{k=1}^{K} \delta_{mn}^k (p_n^k \frac{S_m}{F_n} + r_n^k \frac{S_m}{C_n}). \tag{19}$$

As the only optimization variable contained in D_{mn}^0 is the CNN layer scheduling variable, denoted by δ_{mn}^k, we may design the optimal CNN layer scheduling strategy by minimizing D_{mn}^0. Hence, the optimization problem formulated in (18) is now reduced to the CNN layer scheduling subproblem, which is formulated as

$$\begin{aligned} \min_{\delta_{mn}^k} \ & D_{mn}^0 \\ \text{s.t. } & \text{C1},\ \text{C2},\ \text{C10 in (18)}. \end{aligned} \tag{20}$$

Since the above optimization problem is a simple one-variable optimization problem, we may solve it based on extensive search algorithm and obtain the optimal CNN layer scheduling strategy, denotd by $\delta_{mn}^{k,*}$.

4.2 Task Offloading Subproblem

Substituting $\delta_{mn}^{k,*}$ into the optimization problem formulated in (18), we can observe that the optimization problem is now a joint task offloading and resource allocation problem, and the objective function D can be rewritten as D^0, i.e.,

$$D^0 = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{mn}\left(\frac{\hat{D}_{mn}^{\mathrm{t}}}{\alpha_{mn}} + \frac{\hat{D}_{mn}^{\mathrm{p}}}{\beta_{mn}} + \frac{\hat{D}_{mn}^{\mathrm{tc}}}{\lambda_{mn}}\right) \tag{21}$$

where $\hat{D}_{mn}^{\mathrm{t}} = \frac{S_m}{r_{mn}}$, $\hat{D}_{mn}^{\mathrm{p}} = \sum_{k=1}^{K} \delta_{mn}^{k,*} \frac{p_n^k S_m}{F_n}$, $\hat{D}_{mn}^{\mathrm{tc}} = \sum_{k=1}^{K} \delta_{mn}^{k,*} \frac{r_n^k S_m}{C_n}$.

It can be observed that the objective function D^0 is a mixed discrete and second order function of the optimization variables x_{mn}, α_{mn}, β_{mn} and λ_{mn}, which is notoriously difficult to solve. To address the difficulties, we apply variable transformation, discrete variable relaxation method and the RLT to reformulate the problem.

To tackle the problem of fraction optimization, we first define $\iota_{mn} = \frac{1}{\alpha_{mn}+\varepsilon_b}$, $\nu_{mn} = \frac{1}{\beta_{mn}+\varepsilon_c}$, and $\emptyset_{mn} = \frac{1}{\lambda_{mn}+\varepsilon_f}$ where ε_b, ε_c and ε_f are microscales introduced to avoid divide-by-zero error, then we can rewrite the optimization problem in (18) as follows.

$$\begin{aligned}
&\min_{\substack{x_{mn}\\ \iota_{mn},\nu_{mn},\emptyset_{mn}}} \sum_{m=1}^{M}\sum_{n=1}^{N} x_{mn}(\iota_{mn}\hat{D}_{mn}^{\mathrm{t}} + \nu_{mn}\hat{D}_{mn}^{p} + \emptyset_{mn}\hat{D}_{mn}^{tc}) \\
&\text{s.t.} \quad \text{C3, C4, C9, C10 in (18)} \\
&\quad \text{C11}: \sum_{m=1}^{M}\frac{1}{\iota_{mn}} \le 1 + M\varepsilon_b, \iota_{mn} \in \left[\frac{1}{1+\varepsilon_b}, \frac{1}{\varepsilon_b}\right] \\
&\quad \text{C12}: \sum_{m=1}^{M}\frac{1}{\nu_{mn}} \le 1 + M\varepsilon_c, \nu_{mn} \in \left[\frac{1}{1+\varepsilon_c}, \frac{1}{\varepsilon_c}\right] \\
&\quad \text{C13}: \sum_{m=1}^{M}\frac{1}{\emptyset_{mn}} \le 1 + M\varepsilon_f, \emptyset_{mn} \in \left[\frac{1}{1+\varepsilon_f}, \frac{1}{\varepsilon_f}\right].
\end{aligned} \tag{22}$$

Problem (22) is still a non-convex problem because of the discrete variable x_{mn} and the second order form of the optimization variables. We now employ discrete variable relaxation method to convert $x_{mn} \in \{0,1\}$ to $0 \le x_{mn} \le 1$, then we adopt the RLT to linearize the objective function and constraints in (22). To linearize the second order terms $x_{mn}\iota_{mn}$, $x_{mn}\nu_{mn}$ and $x_{mn}\emptyset_{mn}$, we define $\varphi_{mn} = x_{mn}\iota_{mn}$, $\varsigma_{mn} = x_{mn}\nu_{mn}$ and $\vartheta_{mn} = x_{mn}\emptyset_{mn}$. Considering the constraints on x_{mn}, ι_{mn}, ν_{mn} and $\emptyset_{mn}$, we can obtain respectively the RLT bound-factor product constraints for φ_{mn}, ς_{mn} and ϑ_{mn} as

$$\Xi_{mn}^{\varphi} = \begin{cases} \varphi_{mn} - \frac{1}{1+\varepsilon_b}x_{mn} \ge 0 \\ \iota_{mn} - \frac{1}{1+\varepsilon_b} - \varphi_{mn} + \frac{1}{1+\varepsilon_b}x_{mn} \ge 0 \\ \frac{1}{\varepsilon_b}x_{mn} - \varphi_{mn} \ge 0 \\ \frac{1}{\varepsilon_b} - \iota_{mn} - \frac{1}{\varepsilon_b}x_{mn} + \varphi_{mn} \ge 0 \end{cases} \tag{23}$$

$$\Xi_{mn}^{\varsigma} = \begin{cases} \varsigma_{mn} - \frac{1}{1+\varepsilon_c} x_{mn} \geq 0 \\ \nu_{mn} - \frac{1}{1+\varepsilon_c} - \varsigma_{mn} + \frac{1}{1+\varepsilon_c} x_{mn} \geq 0 \\ \frac{1}{\varepsilon_c} x_{mn} - \varsigma_{mn} \geq 0 \\ \frac{1}{\varepsilon_c} - \nu_{mn} - \frac{1}{\varepsilon_c} x_{mn} + \varsigma_{mn} \geq 0 \end{cases} \tag{24}$$

$$\Xi_{mn}^{\vartheta} = \begin{cases} \vartheta_{mn} - \frac{1}{1+\varepsilon_f} x_{mn} \geq 0 \\ \varnothing_{mn} - \frac{1}{1+\varepsilon_f} - \vartheta_{mn} + \frac{1}{1+\varepsilon_f} x_{mn} \geq 0 \\ \frac{1}{\varepsilon_f} x_{mn} - \vartheta_{mn} \geq 0 \\ \frac{1}{\varepsilon_f} - \varnothing_{mn} - \frac{1}{\varepsilon_f} x_{mn} + \vartheta_{mn} \geq 0 \end{cases} \tag{25}$$

After substituting φ_{mn}, ς_{mn} and ϑ_{mn} into (22), we obtain a convex optimization problem:

$$\begin{aligned} \min_{\substack{x_{mn} \\ \iota_{mn},\nu_{mn},\varnothing_{mn} \\ \varphi_{mn},\varsigma_{mn},\vartheta_{mn}}} & \sum_{m=1}^{M} \sum_{n=1}^{N} (\varphi_{mn} \hat{D}_{mn}^{t} + \varsigma_{mn} \hat{D}_{mn}^{p} + \vartheta_{mn} \hat{D}_{mn}^{tc}) \\ \text{s.t.} \quad & \text{C4, C9, C10 in (18)} \\ & \text{C11}-\text{C13 in (22)} \\ & \text{C14} : 0 \leq x_{mn} \leq 1 \\ & \text{C15} : \varphi_{mn} \in \Xi_{mn}^{\varphi} \\ & \text{C16} : \varsigma_{mn} \in \Xi_{mn}^{\varsigma} \\ & \text{C17} : \vartheta_{mn} \in \Xi_{mn}^{\vartheta} \end{aligned} \tag{26}$$

Problem (26) is now a convex optimization problem, therefore it can be solved efficiently in polynomial time using standard software such as CVX, MOSEK, etc. Let $\left\{\hat{x}_{mn},\ \hat{\iota}_{mn},\ \hat{\nu}_{mn},\ \hat{\varnothing}_{mn},\ \hat{\varphi}_{mn},\ \hat{\varsigma}_{mn},\ \hat{\vartheta}_{mn}\right\}$ denote the optimal solution of (26). As $\hat{x}_{mn}$ is a continuous approximation of x_{mn}, to obtain the binary task offloading strategy x_{mn}^{*} of (18), we define

$$\begin{cases} x_{mn}^{*} = 1, & \hat{x}_{mn} \geq 0.5 \\ x_{mn}^{*} = 0, & \hat{x}_{mn} < 0.5. \end{cases} \tag{27}$$

4.3 Resource Allocation Subproblem

While the optimal resource allocation strategy denoted by $\hat{\alpha}_{mn}$, $\hat{\beta}_{mn}$ and $\hat{\lambda}_{mn}$ can be obtained from $\hat{\iota}_{mn}$, $\hat{\nu}_{mn}$, $\hat{\varnothing}_{mn}$, the suboptimality may occur due to the approximation of x_{mn}. In this paper, given the task offloading strategy x_{mn}^{*}, we further formulate the resource allocation subproblem and calculate the optimal resource allocation strategy by means of Lagrange dual method.

Based on the obtained task offloading strategy x_{mn}^{*}, we calculate the number of tasks being offloaded to individual MEC servers. In the case that $\sum_{m=1}^{M} x_{mn}^{*} \geq 1$ for any E_n, i.e., more than one task is offloaded to E_n, resource sharing occurs

at E_n and the optimal resource allocation strategy should be designed. Let $\Phi_n = \{\mathrm{T}_m | x^*_{mn} = 1\}$ denote the set of MD tasks which are offloaded to E_n.

We denote $\bar{D}_{mn}$ as the task latency when T_m is offloaded to E_n, i.e., $x^*_{mn} = 1$, we obtain

$$\bar{D}_{mn} = \frac{S_m}{R_{mn}} + \sum_{k=1}^{K} \delta^{k,*}_{mn} \left(\frac{p^k_n S_m}{\beta_{mn} F_n} + \frac{r^k_n S_m}{\lambda_{mn} C_n} \right). \tag{28}$$

The resource allocation subproblem can be formulated as

$$\begin{aligned} \min_{\alpha_{mn},\beta_{mn},\lambda_{mn}} \quad & \sum_{\mathrm{T}_m \in \Phi_n} \bar{D}_{mn} \\ \text{s.t.} \quad & \mathrm{C5} - \mathrm{C10}. \end{aligned} \tag{29}$$

It can be proved that the optimization problem formulated in (29) is a convex problem which can be solved by using the Lagrange dual method. The corresponding Lagrange function can be expressed as

$$\begin{aligned} & L\left(\alpha_{mn}, \beta_{mn}, \lambda_{mn}, \eta_{mn}, \mu_{mn}, \gamma_{mn}, \theta_{mn}, \omega_{mn}, \varepsilon, \tau, \psi\right) \\ & = \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \bar{D}_{mn} + \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \eta_{mn} \left(\bar{D}_{mn} - D^{\max}_m\right) \\ & + \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \mu_{mn} \left(R^{\min}_m - R_{mn}\right) + \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \gamma_{mn} \left(\alpha_{mn} - 1\right) \\ & + \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \theta_{mn} \left(\beta_{mn} - 1\right) + \sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \omega_{mn} \left(\lambda_{mn} - 1\right) \\ & + \varepsilon \Big(\sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \alpha_{mn} - 1 \Big) + \tau \Big(\sum_{\mathrm{T}_m \in \Phi^n_\mathrm{T}} \beta_{mn} - 1 \Big) \\ & + \psi \Big(\sum_{\mathrm{T}_m \in \Phi_\mathrm{n}} \lambda_{mn} - 1 \Big) \end{aligned} \tag{30}$$

where $\eta_{mn}, \mu_{mn}, \gamma_{mn}, \theta_{mn}, \omega_{mn}, \varepsilon, \tau, \psi$ are Lagrange multipliers. The Lagrange dual problem is formulated as

$$\begin{aligned} \max_{\substack{\eta_{mn},\mu_{mn} \\ \gamma_{mn},\theta_{mn},\omega_{mn} \\ \varepsilon,\tau,\psi}} \quad & \min_{\alpha_{mn},\beta_{mn},\lambda_{mn}} \quad L \\ \text{s.t.} \quad & \eta_{mn}, \mu_{mn}, \gamma_{mn}, \theta_{mn}, \omega_{mn}, \varepsilon, \tau, \psi \geq 0. \end{aligned} \tag{31}$$

For a given set of Lagrange multipliers, the optimal resource allocation strategy can be obtained as

$$\alpha^*_{mn} = \left[\sqrt{\frac{\left(1 + \eta_{mn}\right) S_m}{\left(\gamma_{mn} + \varepsilon - \mu_{mn}\right) r_{mn}}} \right]^+, \tag{32}$$

$$\beta^*_{mn} = \left[\sqrt{\frac{(1 + \eta_{mn}) S_m p^k_n}{\left(\theta_{mn} + \tau\right) F_n}} \right]^+, \tag{33}$$

$$\lambda_{mn}^{*} = \left[\sqrt{\frac{(1+\eta_{mn}) S_m r_n^k}{(\omega_{mn}+\psi) C_n}} \right]^{+} \tag{34}$$

where $[x]^{+} = \max\{x, 0\}$.

Replacing α_{mn}, β_{mn} and λ_{mn} by α_{mn}^{*}, β_{mn}^{*} and λ_{mn}^{*} respectively in D_{mn}, we will be able to obtain the optimal task latency of T_m at E_n when sharing bandwidth, computation and fronthual resource of E_n with other tasks.

5 Simulation Results

In this section, we evaluate the performance of the proposed algorithm by simulations. In the simulation, we consider a rectangular region with the size being 100 m × 100 m where the MEC servers and the MDs are randomly located. The number of MEC servers is set as 2, 3, 4, respectively and the number of tasks is set from 25 to 35, the total number of CNN layers is set as $K = 5$ in the simulation. Other simulation parameters employed in the simulations, unless otherwise mentioned, are summarized in Table 1. The simulation results are averaged over 1000 independent experiments.

Table 1. Simulation parameter

Parameters	Value
Input data size (S_m)	$[1, 10]$Mb
Transmission power of MD_m (P_m)	0.6 W
Noise power (σ^2)	−110 dBm
Channel path loss model	$128.1 + 27\log(d)$dB
Bandwidth of E_n (B_n)	10 MHz
Computation capability of E_n (F_n)	$\{600, 800\}$GHz
Fronthaul capacity of E_n (C_n)	100 Mbps
Reduction ratio (r_n^k)	$[0.4, 0.2, 0.16, 0.128, 0.1152]$
Computational overhead of unit data (p_n^k)	$[0.5, 1.1, 1.8, 2.84, 3.992]$GHz/M

In Fig. 2, we examine the performance of our proposed algorithm, the RLT algorithm without the Lagrangian dual method (RLT-0), the algorithm proposed in [9], the average resource allocation (ARA) algorithm, which allocates the resource of the MEC servers equally to the associated MDs and a benchmark algorithm, i.e., the minimum distance based task offloading (MDO) algorithm, which allows the MDs to select the nearest MEC server to offload their tasks.

The overall task latency obtained from different algorithms is shown in Fig. 2. From the figure, we can observe that while the overall task latency increases as the number of tasks of MDs increases for all the considered algorithms, our proposed algorithm offers the minimum overall task latency compared with other

algorithms. This is mainly benefited from the joint optimization of task offloading, CNN layer scheduling and resource allocation, and the combination of RLT and Lagrange dual method.

In Fig. 3, we plot the overall task latency versus the number of tasks obtained from our proposed algorithm and the one proposed in [9]. Different number of MEC servers, i.e., $N = 2, 3, 4$ is considered in the simulation. From the figure, we can see that lower overall task latency can be achieved with the increased number of the MEC servers. Comparing the results obtained from our proposed algorithm and the one proposed in [9], we can see that our proposed algorithm offers lower overall task latency. The reason is that our proposed algorithm aims to minimize the overall task latency while the authors in [9] tend to minimize the energy consumption required for task execution, hence may result in longer task latency.

In Fig. 4, we examine the overall task latency versus the available bandwidth of the MEC servers. The number of MEC servers and tasks are set as 2 and 30, respectively in the simulation. It can be seen from the figure that the overall task latency decreases as the bandwidth of the MEC servers increases. Comparing the results obtained from the proposed algorithm and the algorithm proposed in [9], we can see that the proposed algorithm offers better performance.

In Fig. 5, we compare the simulation results of our proposed algorithm with two baseline algorithms which employ different task offloading schemes and no cooperative computing is applied. For baseline algorithm 1, we assume that tasks can only be offloaded to the MEC servers and no MCC server is available. For baseline algorithm 2, we assume that no MEC servers are deployed and tasks can only be offloaded to the MCC server. From Fig. 5, we can observe that our proposed algorithm outperforms the two baseline algorithms demonstrating the performance benefits by conducting cooperative computing between the MEC servers and the MCC server.

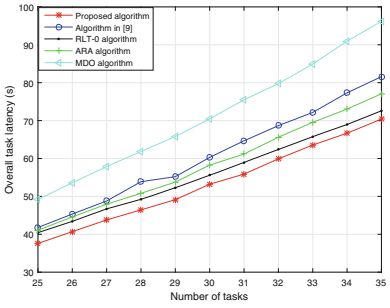

Fig. 2. Overall task latency versus number of tasks (different algorithms).

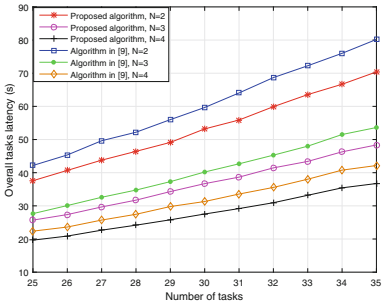

Fig. 3. Overall task latency versus number of tasks (different number of MEC servers).

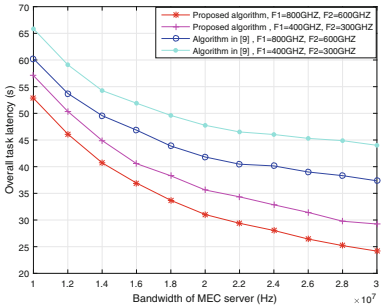

Fig. 4. Overall task latency versus bandwidth of MEC servers.

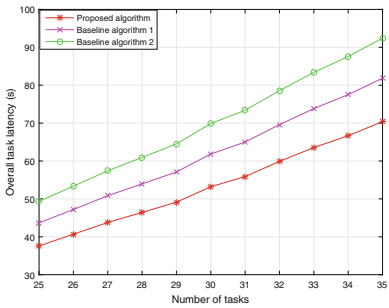

Fig. 5. Overall task latency versus number of tasks (different task offloading schemes).

6 Conclusions

In this paper, we consider a cooperative computing system and formulate the joint task offloading, CNN layer scheduling and resource allocation as an optimization problem which minimizes overall task latency. As the formulated optimization problem is NP-hard, we decompose the original optimization problem into CNN layer scheduling subproblem, task offloading subproblem and resource allocation subproblem, and solve the three subproblems based on extensive search algorithm, the RLT and Lagrangian dual method, respectively. Numerical results demonstrate the proposed algorithm outperforms previously proposed algorithms and the baseline algorithms which fail to apply cooperative computing schemes.

References

1. Fu, X., Secci, S., Huang, D., Jana, R.: Mobile cloud computing. IEEE Commun. Mag. **53**, 61–62 (2015)
2. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
3. Yaseen, M.U., Anjum, A., Antonopoulos, N.: Modeling and analysis of a deep learning pipeline for cloud based video analytics. In: Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing Applications and Technologies, pp. 121–130 (2017)
4. Li, L., Ota, K., Dong, M.: Eyes in the dark: distributed scene understanding for disaster management. IEEE Trans. Parallel Distrib. Syst. (2017). https://doi.org/10.1109/TPDS.2017.2740294
5. Chen, X., Lei, J., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile edge cloud computing. IEEE/ACM Trans. Netw. **24**(5), 2795–2808 (2016)
6. You, C., Huang, K., Chae, H., Kim, B.H.: Energy-efficient resource allocation for mobile edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2017)

7. Dinh, T.Q., Tang, J., La, Q.D., Quek, Q.S.: Offloading in mobile edge computing: task allocation and computational frequency scaling. IEEE Trans. Commun. **65**(8), 4798–4810 (2017)
8. Li, Q., Lei, J., Lin, J.: Min-max latency optimization for multiuser computation offloading in fog-radio access networks (2018)
9. Cheng, K., Teng, Y., Sun, W., Liu, A., Wang, X.: Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems. In: Proceedings of IEEE International Conference on Communications (ICC 2018), July 2018
10. Li, H., Ota, K., Dong, M.: Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Netw. **32**(1), 96–101 (2018)