Joint Resource Allocation for Latency-Sensitive Services over Mobile Edge Computing Networks with Caching

Article · October 2018

DOI: 10.1109/JIOT.2018.2875917

CITATIONS
3

READS
290

9 authors, including:

Jiao Zhang
National University of Defense Technology
9 PUBLICATIONS **72** CITATIONS

SEE PROFILE

Zhaolong Ning
Dalian University of Technology
122 PUBLICATIONS **1,437** CITATIONS

SEE PROFILE

Li Zhou
National University of Defense Technology
38 PUBLICATIONS **231** CITATIONS

SEE PROFILE

Jun Cheng
Chinese Academy of Sciences
156 PUBLICATIONS **1,482** CITATIONS

SEE PROFILE

Joint Resource Allocation for Latency-Sensitive Services over Mobile Edge Computing Networks with Caching

Jiao Zhang, Xiping Hu, Zhaolong Ning, Edith C.-H. Ngai, Li Zhou, Jibo Wei, Jun Cheng, Bin Hu, Victor C.M. Leung

***Abstract*—Mobile edge computing (MEC) has risen as a promising paradigm to provide high quality of experience via relocating the cloud server in close proximity to smart mobile devices (SMDs). In MEC networks, the MEC server with computation capability and storage resource can jointly execute the latency-sensitive offloading tasks and cache the contents requested by SMDs. In order to minimize the total latency consumption of the computation tasks, we jointly consider computation offloading, content caching and resource allocation as an integrated model, which is formulated as a mix integer nonlinear programming (MINLP) problem. We design an asymmetric search tree and improve the branch and bound method to obtain a set of accurate decisions and resource allocation strategies. Furthermore, we introduce the auxiliary variables to reformulate the proposed model and apply the modified generalized benders decomposition method to solve the MINLP problem in polynomial computation complexity time. Simulation results demonstrate the superiority of the proposed schemes.**

***Index Terms*—Mobile edge computing; Content caching; Resource allocation; Internet of Things**

I. Introduction

With the rapid development of 5G network and Internet of Things (IoT), computation-intensive applications, such as natural language processing, face recognition and augmented reality, are explosively emerging [1]. Smart mobile devices (SMDs) as the major application supporting platform, have the limited computation capability (e.g., central process unit (CPU) cycle frequency) and storage resource. To address the conflict of the resource-intensive applications and limited capability of SMDs, offloading the computation tasks to the powerful cloud server by mobile cloud computing is an available alternative approach [2]. However, due to the infrastructure-based cloud servers are far from SMDs, the long transmissions are inevitable and challenge the quality of experience (QoE), especially for the latency-sensitive applications.

Mobile edge computing (MEC) is envisioned as a promising technology to react the above problem by deploying the cloud server at the edge of the mobile networks, in proximity to SMDs [3], [4]. If the MEC server is equipped with computation capability and storage resource, computation offloading and content caching approaches can be jointly employed to improve user experience, especially network latency. Computation offloading migrates the computation tasks from SMDs to the resourceful MEC server, so that the task execution can be accelerated. However, offloading overmuch computation tasks will invoke extra overhead for MEC networks under the constraints of communication and computation resources. Consequently, offloading decision, i.e., whether to execute the computation task on the MEC server or locally, becomes an important issue for each SMD. In [5]–[8], the problem has been investigated for the purpose of improving the system utility, such as the reduction of the total latency consumption, the minimum energy consumption or the trade-off between latency and energy. Furthermore, content caching means that the requested contents can be cached from the Internet to the MEC server. This process can not only avoid duplicate transmissions of the same content, but also alleviate the bandwidth of backhaul link [9], [10]. Nevertheless, due to the limited storage space of the MEC server, the bottleneck of caching is expected as the increasing number of the requested contents. Efficient caching strategies can be applied upon different contents. A large number of efforts on caching strategies have been developed in the web and the information-centric networks [11], [12]. What's more, the integration of computation offloading and content caching has been attracting more attentions with the emergence of MEC networks [13]–[15].

In this paper, we investigate a centralized MEC framework with computation capability and content caching, where spectrum and computation resources are jointly considered to satisfy the requirements of latency sensitive computation tasks. Our designed framework aims to minimize the total latency consumption of all the computation tasks in the whole network. The main contributions of this paper are summarized as follows:

- We jointly consider the computation offloading, content caching and resource allocation to formulate an optimiza-

J. Zhang, L. Zhou (Co-corresponding author) and J. Wei are with the College of Electronic Science, National University of Defense Technology, Changsha, China. E-mail: {zhangjiao16,zhouli2035,wjbhw}@nudt.edu.cn.

J. Zhang, X. Hu (Co-corresponding author) and J. Cheng (Co-corresponding author) are with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. E-mail: {xp.hu,jun.cheng}@siat.ac.cn.

Z. Ning (Co-corresponding author) is with the School of Software, Dalian University of Technology, Dalian, China. E-mail: zhaolongning@dlut.edu.cn.

E. Ngai is with the Department of Information Technology, Uppsala University, Uppsala, Sweden. Email: edith.ngai@it.uu.se.

B. Hu (Co-corresponding author) is with School of Information Science and Engineering, Lanzhou University, China. Email: bh@lzu.edu.cn.

Victor C.M. Leung is with Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. Email: vleung@ece.ubc.ca.

tion problem to improve the quality of user experience for SMDs with latency-sensitive tasks.
- Based on the branch and bound method and problem constraints, we modify the deep search tree and adopt the improved branch and bound method to solve the optimization problem.
- We decompose the original non-convex problem and apply the modified generalized benders decomposition method to solve the problem with polynomial computation complexity.

The rest of this paper is organized as follows. In section II, we review related works. The system model and problem formulation are described in Section III. Section IV specifies the problem reformulation and solution, where the modified branch and bound method together with the generalized benders decomposition are presented. Simulation results are discussed in section V. Conclusions are drawn in Section VI.

II. Related Works

MEC-enabled networks equipped with cloud computing and caching capabilities contribute to the improvement of user experience compared with mobile cloud computing. However, due to the competition for the limited resources of computation, communication and storage, computation offloading or content caching strategy is always investigated coupled with resource optimization. The authors in [6] considered an energy-saving offloading framework by jointly optimizing radio and computation resources in MEC networks. The authors in [16] studied the problem of mobile edge computation offloading in ultra-dense IoT networks, so as to minimize the overall computation overhead of all tasks while satisfying the channel constraint. In [17], a semi-distributed heuristic offloading decision algorithm was proposed to maximize the system utility for the constrained transmission power and computation resources. An optimal cooperative content caching and delivery policy was proposed in [10] to minimize the average downloading latency under the constraints of storage and bandwidth capabilities. In [15], a joint caching and processing problem was formulated, subjecting to the caching and processing capacity constraints. Most of these works are focusing on improving the performance of MEC networks by utilizing computation offloading, resource allocation or content caching strategies. However, these strategies are generally adopted separately.

Some few works have considered the integration of computation offloading, content caching and resource allocation. In [18], an integrated problem of computation offloading, content caching and resource allocation was formulated to maximize the total revenue of MEC system operations. In the full duplex-enabled small cell network with MEC and caching, a virtual resource allocation was investigated in terms of QoE of users and the corresponding resource consumption [19]. However, both [18] and [19] aim to maximize the total system revenue, such as the saving radio resource, computation resource and backhaul bandwidth, etc. When it comes to the specific optimal objective (e.g., total latency or energy consumption), the proposed schemes are not efficient. A computation resource management scheme was studied by genetic algorithm to enhance mobile terminal performance through integrating computation offloading and data caching [20]. However, there is no consideration about communication resource allocation. In [21], the authors established the joint problem of caching, computing and bandwidth resource allocation, and the objective is to minimize the combination cost of network usage and energy consumption. The interference alignment problem was investigated in the communications, caching and computing oriented small cell networks [22].

Different from previous works, in this paper, we jointly consider computation offloading, content caching, communication and computation resource allocation in a centralized MEC network, aiming to minimize the specific total latency consumption.

III. System Model and Problem Formulation

An illustrative system model of a centralized MEC network is shown in Fig. 1. It consists of one macro eNodeB (MeNB) and N SMDs. The MeNB equipped with an MEC server is capable of executing computation-intensive tasks, which is connected to the Internet via the core network of cellular communication systems. SMDs are associated with the MeNB using Orthogonal Frequency-Division multiplexing (OFDM), where the spectrum resource allocated to each SMD cannot be reused. The set of SMDs overlaid by MeNB is denoted by $\mathcal{N} = \{1, 2, ..., N\}$. In the network, SMD i has a computation task $\tau_i = \{d_i, e_i, T_i^{\max}\}$ to be executed, where d_i denotes the input data size of the computation task, e_i stands for the number of CPU cycles required for task execution, and $T_i^{\max}$ is the maximum tolerance latency. In order to accomplish the computation tasks, the MEC server can request contents (e.g., program codes) from the Internet through the backhaul link. The MEC server has a content caching container and can decide whether to store the content or not. If the content has been stored, it can be reused. For each SMD, its task can be either computed locally or offloaded to the MEC server for execution.

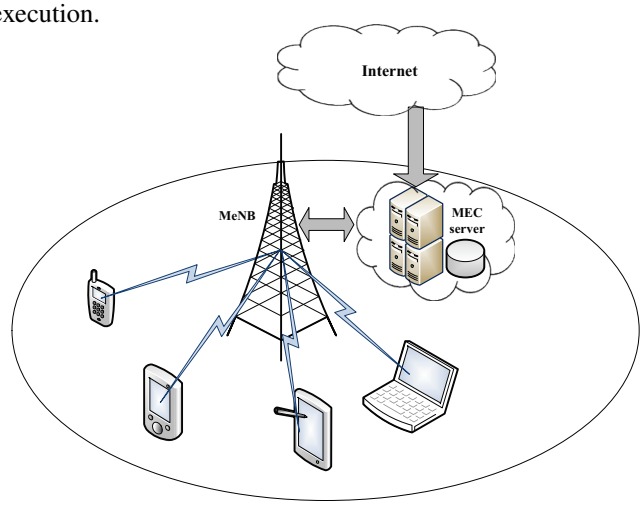

Fig. 1: An illustrative network model.

We define the computation offloading decision of SMD i as s_i, and $s_i \in \{0, 1\}$. $s_i = 1$ means the computation task is

remotely computed on the MEC server, while we set $s_i = 0$ for local computation. The notations used in this paper are summarized in Table I.

TABLE I: Notations.

Notation	Description
$\mathcal{N}$	The set of SMDs
s_i	Computation offloading decision of SMD i
c_i	Content caching decision of SMD i
a_i	The percentage of spectrum resource allocated to SMD i
b_i	The percentage of computation resource allocated to SMD i
p_i	The transmission power of SMD i
h_i	The channel gain between SMD and MeNB
δ^2	Thermal noise power
B	Total bandwidth
F	Total computation capability of the MEC server
f_i^l	Computation capability of SMD i
M	Total storage capability of the MEC server
$p_i(j)$	The popularity of the jth content requested by SMD i
R_i^b	The alleviated backhaul bandwidth

A. Communication model

When SMD i offloads the computation task to the MEC server, the input data of the task needs to be transmitted to the MEC server. For the uplink transmission, the achievable spectrum efficiency of SMD i is denoted as

$$r_i = \log_2(1 + \frac{p_i h_i}{\sigma^2}), \tag{1}$$

where p_i is the transmission power of SMD, and h_i represents the channel gain between SMD and MeNB. σ^2 denotes the background noise power, and we assume that it is constant with respect to the slow fading channel [23].

The whole available spectrum bandwidth $B\ Hz$ can be divided and assigned to SMDs for input data transmission. Let $a_i \in [0, 1]$ denote the percentage of the spectrum bandwidth allocated to SMD i by the MeNB, the uplink transmission rate of SMD i can be calculated by

$$R_i^u = a_i B r_i. \tag{2}$$

Hence, the uplink transmission time depends on the size of the input data, and the uplink transmission rate is

$$t_i^u = \frac{d_i}{R_i^u}. \tag{3}$$

B. Computation model

The task of SMD can be computed locally or remotely on the MEC server via computation offloading. The corresponding computation overhead (e.g., computation time) is mainly relevant to the required CPU cycles of the computation task.

1) Local computing

Let f_i^l denote the local computation capability (i.e., CPU cycles per second) of SMD i. If SMD executes its task locally, the local computation time can be obtained by

$$t_i^L = \frac{e_i}{f_i^l}. \tag{4}$$

2) Edge computing

The total computation capability of the MEC server is represented as F. $b_i \in [0, 1]$ is the percentage of computation resource allocated to SMD i by the MEC server. The edge computation time of the task offloaded to the MEC server can be described as

$$t_i^e = \frac{e_i}{b_i F}. \tag{5}$$

C. Caching model

We denote $c_i \in \{0, 1\}$ as the caching decision for SMD i. $c_i = 1$ indicates that the MEC server chooses to cache the content requested by SMD i, and $c_i = 0$ otherwise.

If the task is offloaded to the MEC server, the server can request the content from the Internet. Based on [24] and [25], we assume that the popularity of the requested contents is modeled as Zipf distribution. Thus, the popularity of the jth popular content requested by SMD i can be given by

$$p_i(j) = \frac{1/j^\alpha}{\sum_{j=1}^{N_f} 1/j^\alpha}, \tag{6}$$

where N_f is the total types of contents in the Internet. α represents a shape parameter of Zipf distribution, which can be set as a constant value 0.56 [18], [26].

According to [18] and [27], if the requested content is cached, the system will be rewarded with the reduction of backhaul delay or the alleviation of backhaul bandwidth between MeNB and the Internet. The alleviated backhaul bandwidth for caching jth popular content requested by SMD i can be obtained by

$$R_i^b = p_i(j)\bar{R}, \tag{7}$$

where $\bar{R}$ is the average data transmission rate in the Internet. From (7), it can be seen that caching performance depends crucially on the popularity of requested contents. If the size of the content requested by SMD i is known as f_i, then the reduced backhaul delay can be derived as

$$t_i^c = \frac{f_i}{R_i^b}, \tag{8}$$

Note that the caching capability of the MEC server is constrained as M. It is not possible to cache all contents. Besides, SMDs have no ability of caching. Assume that the application programs downloaded by SMDs have all basic program codes to execute tasks. Hence, the content delivery is ignored for local computing.

Therefore, when the task is executed by edge computing, the total latency consumption of the offloading task is comprised of the transmission time, execution time and the backhaul delay. It can be calculated by

$$t_i^M = t_i^u + t_i^e + (1 - c_i)t_i^c. \tag{9}$$

where the backhaul delay can be removed for $c_i = 1$, due to the fact that the MEC server can adopt the cached content to compute the task directly. Otherwise, the MEC server needs to request the content from the Internet. Similar to [7] and [18],

the time overhead of computation outcome transmitted from the MEC server to SMD is ignored in our system.

For SMD i, no matter whether its task is executed by local computing or edge computing, the total latency of the task can be represented by

$$T_i = s_i t_i^M + (1 - s_i) t_i^L. \tag{10}$$

D. Problem formulation

By jointly considering computation offloading, caching decision and resource allocation, this paper aims to minimize the total latency for fulfilling the computation tasks in MEC networks. The problem model can be formulated as follows:

$$\begin{aligned}
P0: \quad & \min_{\mathbf{s},\mathbf{c},\mathbf{a},\mathbf{b}} \sum_{i=1}^{N} T_i \\
s.t. \quad & C1: 0 \leqslant a_i \leqslant s_i, \forall i \\
& C2: \sum_{i=1}^{N} a_i \leqslant 1 \\
& C3: 0 \leqslant b_i \leqslant s_i, \forall i \\
& C4: \sum_{i=1}^{N} b_i \leqslant 1 \\
& C5: 0 \leqslant c_i \leqslant s_i, \forall i \\
& C6: \sum_{i=1}^{N} c_i f_i \leqslant M \\
& C7: s_i\left(\frac{d_i}{a_i B r_i} + \frac{e_i}{b_i F} + (1 - c_i)\frac{f_i}{R_i^b}\right) + (1 - s_i)\frac{e_i}{f_i^l} \\
& \quad \leqslant T_i^{\max}, \forall i \\
& C8: s_i \in \{0, 1\}, \forall i \\
& C9: c_i \in \{0, 1\}, \forall i
\end{aligned} \tag{11}$$

where $C1$ and $C3$ are the constraints of available spectrum and computation resources allocated to SMD i, respectively. $C2$ indicates that the sum of spectrum resources allocated to SMDs with offloading tasks cannot exceed the total available spectrum bandwidth. $C4$ means the sum of the allocated computation resources for offloading tasks cannot exceed the total computation capability of the MEC server. $C5$ guarantees that only SMDs with offloading tasks need to request the contents from the Internet. $C6$ ensures that the sum of the contents cached from the Internet cannot exceed the storage capability of the MEC server. $C7$ addresses that the total latency of each computation task should satisfy the maximum tolerance latency. $C8$ and $C9$ are the binary variables to represent offloading decision and caching decision, respectively.

In addition, we assume that $e_i/f_i^l \leqslant T_i^{\max}$ is always satisfied, so as to guarantee that problem $P0$ has an optimal solution [6].

IV. Problem Transformation and Solution

In this section, we first present some problem observation and reformulation. Then, the modified branch and bound method (MBB) and the modified general benders decomposition (MGBD) are introduced to solve the optimization problem.

From the observation of problem $P0$, we find that both offloading decision $\mathbf{s}$ and caching decision $\mathbf{c}$ are integer vectors, and resource allocation vectors of $\mathbf{a}$ and $\mathbf{b}$ are continuous. Meanwhile, the integer vectors are coupled with the continuous vectors. Therefore, problem $P0$ is a mixed integer nonlinear programming [28], which is intractable to find the optimal solution.

However, if the integer vectors of $\mathbf{s}$ and $\mathbf{c}$ can be determined, namely the sets of $\mathcal{N}_1^s = \{i | s_i = 1, i \in \mathcal{N}\}$, $\mathcal{N}_0^s = \{i | s_i = 0, i \in \mathcal{N}\}$, $\mathcal{N}_1^c = \{i | c_i = 1, i \in \mathcal{N}\}$ and $\mathcal{N}_0^c = \{i | c_i = 0, i \in \mathcal{N}\}$ are available, then problem $P0$ can be converted as problem $P1$:

$$\begin{aligned}
P1: \quad & \min_{\mathbf{a},\mathbf{b}} \sum_{i \in \mathcal{N}_1^s} \left(\frac{d_i}{a_i B r_i} + \frac{e_i}{b_i F} + (1 - c_i)\frac{f_i}{R_i^b}\right) + \sum_{i \in \mathcal{N}_0^s} \frac{e_i}{f_i^l} \\
s.t. \quad & C1, C2, C3, C4, \forall i \in \mathcal{N} \\
& C6: \sum_{i \in \mathcal{N}_1^c} f_i \leqslant M \\
& C7: \frac{d_i}{a_i B r_i} + \frac{e_i}{b_i F} + (1 - c_i)\frac{f_i}{R_i^b} \leqslant T_i^{\max}, \forall i \in \mathcal{N}_1^s
\end{aligned} \tag{12}$$

where offloading decision and caching decision vectors are decoupled from resource allocation vectors. Obviously, problem $P1$ is a convex optimization problem for the fixed integer variables [29].

A. Modified branch and bound method

According to the special constitution of problem $P1$, we can adopt the branch and bound method [30], [31] for the original problem $P0$, which is sophisticated for the MINLP problem. The idea of the branch and bound method is similar to exhaustive search. It uses all integer variables to establish a completed search tree and employs the depth-first search strategy to find the accurate optimal solution. In our model, the integer variables $\{0, 1\}$ are the set of offloading decision and caching decision, which can be represented as $(\mathbf{s}, \mathbf{c})$. Furthermore, by considering the constraint of offloading decision and caching decision ($C5$) in problem $P0$, we establish an asymmetric search tree, as shown in Fig. 2.

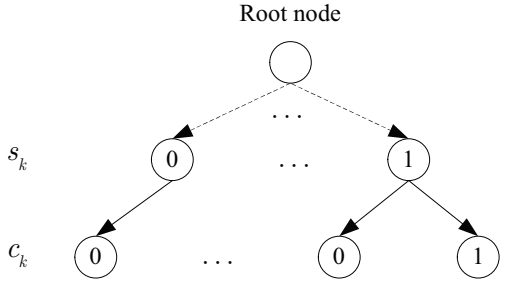

Fig. 2: Search tree of the MBB method.

In the asymmetric search tree, the root node of the tree represents the original problem $P0$. The determination of offloading decision always precedes caching decision. We assume that offloading decision s_k and caching decision c_k are located in the lth and $(l+1)$th layers of the search tree, respectively. Different from the completed and symmetric tree, there is only one child node $c_k = 0$ when the value of s_k is determined as 0 in the tree. If the father node s_k is determined as 1, it has two child nodes, i.e., the left child node $c_k = 0$ and the right node $c_k = 1$. By traversing a path from the root node to the leaf node, a group of integer vectors and

the corresponding continuous vectors can be solved. After the whole search tree is traversed, the optimal path and solutions can be determined eventually. More detailed design of the MBB algorithm can refer to [6] and [30].

B. Modified generalized benders decomposition

In order to reduce the computation complexity, the generalized benders decomposition (GBD) method [32], [33] is introduced to solve the formulated problem. The GBD method decomposes the integer vectors and continuous vectors into a master problem and a primal problem, respectively. The master problem is a mixed integer linear programming (MILP) and the primal problem is a nonlinear programming (NLP). The optimal solution of the original problem can be obtained by iteratively solving the primal problem and the master problem.

In case of the divide-by-zero error, we introduce two microscales ε_1 and ε_2 to the continuous variables, namely $u_i = a_i + \varepsilon_1$ and $v_i = b_i + \varepsilon_2$, where $u_i \in [\varepsilon_1, 1+\varepsilon_1]$ and $v_i \in [\varepsilon_2, 1+\varepsilon_2]$ should hold. Then, problem $P0$ can be reformulated into problem $P2$.

$$\begin{aligned}
P2: \quad & \min_{\mathbf{s},\mathbf{c},\mathbf{u},\mathbf{v}} \sum_{i=1}^{N} \{s_i(\tfrac{d_i}{u_i B r_i} + \tfrac{e_i}{v_i F} + (1-c_i)\tfrac{f_i}{R_i^b}) + (1-s_i)\tfrac{e_i}{f_i^l}\} \\
s.t. \quad & C1: \varepsilon_1 \leqslant u_i \leqslant s_i + \varepsilon_1, \forall i \\
& C2: \sum_{i=1}^{N} u_i \leqslant 1 + N\varepsilon_1 \\
& C3: \varepsilon_2 \leqslant v_i \leqslant s_i + \varepsilon_2, \forall i \\
& C4: \sum_{i=1}^{N} v_i \leqslant 1 + N\varepsilon_2 \\
& C7: s_i(\frac{d_i}{u_i B r_i} + \frac{e_i}{v_i F} + (1-c_i)\frac{f_i}{R_i^b}) + (1-s_i)\frac{e_i}{f_i^l} \\
& \quad \leqslant T_i^{\max}, \forall i \\
& C5, C6, C8, C9
\end{aligned} \tag{13}$$

According to the definition of GBD, the primal problem $P2.1$ can be obtained by fixing the binary variables in problem $P2$.

$$\begin{aligned}
P2.1: \quad & \min_{\mathbf{u},\mathbf{v}} \sum_{i=1}^{N} \{\bar{s}_i^k(\tfrac{d_i}{u_i B r_i} + \tfrac{e_i}{v_i F} + (1-\bar{c}_i^k)\tfrac{f_i}{R_i^b}) \\
& \quad + (1-\bar{s}_i^k)\frac{e_i}{f_i^l}\} \\
s.t. \quad & C1: \varepsilon_1 \leqslant u_i \leqslant \bar{s}_i^k + \varepsilon_1, \forall i \\
& C2: \sum_{i=1}^{N} u_i \leqslant 1 + N\varepsilon_1 \\
& C3: \varepsilon_2 \leqslant v_i \leqslant \bar{s}_i^k + \varepsilon_2, \forall i \\
& C4: \sum_{i=1}^{N} v_i \leqslant 1 + N\varepsilon_2 \\
& C5: 0 \leqslant \bar{c}_i^k \leqslant \bar{s}_i^k, \forall i \\
& C6: \sum_{i=1}^{N} \bar{c}_i^k f_i \leqslant M \\
& C7: \bar{s}_i^k(\frac{d_i}{u_i B r_i} + \frac{e_i}{v_i F} + (1-\bar{c}_i^k)\frac{f_i}{R_i^b}) + (1-\bar{s}_i^k)\frac{e_i}{f_i^l} \\
& \quad \leqslant T_i^{\max}, \forall i
\end{aligned} \tag{14}$$

For a fixed realization of the binary variables, the primal problem $P2.1$ is convex. If the primal problem is feasible, it has a unique continuous solution of $(\mathbf{u}^k, \mathbf{v}^k)$. The corresponding Lagrange multiplier λ_k for the constraints can be obtained as well. In addition, the optimal objective function value of the primal problem is the valid upper bound (UB) of problem $P2$.

To simplify the description, we adopt $\mathbf{x}$ and $\mathbf{y}$ to represent the set of continuous vectors and the set of binary vectors, namely $\mathbf{x} = (\mathbf{u}, \mathbf{v})$ and $\mathbf{y} = (\mathbf{s}, \mathbf{c})$. The objective function of problem $P2.1$ can be denoted by $F(\mathbf{x}, \mathbf{y})$, which can be described as

$$\begin{aligned}
F(\mathbf{x}, \bar{\mathbf{y}}^k) = \sum_{i=1}^{N} \{ & \bar{y}_i^k(\frac{d_i}{x_i B r_i} + \frac{e_i}{x_{i+N} F} + (1-\bar{y}_{i+N}^k)\frac{f_i}{R_i^b}) \\
& + (1-\bar{y}_i^k)\frac{e_i}{f_i^l}\},
\end{aligned} \tag{15}$$

where $\bar{\mathbf{y}}^k = (\bar{\mathbf{s}}^\mathbf{k}, \bar{\mathbf{c}}^\mathbf{k})$, and it represents the set of fixed integer variables. We define $G(\mathbf{x}, \mathbf{y})$ as the constraint set of the primal problem $P2.1$.

$$G(\mathbf{x}, \bar{\mathbf{y}}^k) = \begin{pmatrix}
x_i - \bar{y}_i^k - \varepsilon_1, \forall i \in \mathcal{N} \\
\sum_{i=1}^{N} x_i - 1 - N\varepsilon_1 \\
x_{i+N} - \bar{y}_i^k - \varepsilon_2, \forall i \in \mathcal{N} \\
\sum_{i=1}^{N} x_{i+N} - 1 - N\varepsilon_2 \\
\bar{y}_{i+N}^k - \bar{y}_i^k, \forall i \in \mathcal{N} \\
\sum_{i=1}^{N} \bar{y}_{i+N}^k f_i - M \\
\bar{y}_i^k(\frac{d_i}{x_i B r_i} + \frac{e_i}{x_{i+N} F} + (1-\bar{y}_{i+N}^k)\frac{f_i}{R_i^b}) \\
+(1-\bar{y}_i^k)\frac{e_i}{f_i^l} - T_i^{\max}, \forall i \in \mathcal{N}
\end{pmatrix}_{(4N+3)\times 1} . \tag{16}$$

Thereafter, problem $P2.1$ can be converted as the following form.

$$\begin{aligned}
P2.1: \quad & \min_{\mathbf{x}} F(\mathbf{x}, \bar{\mathbf{y}}^k) \\
s.t. \quad & G(\mathbf{x}, \bar{\mathbf{y}}^k) \leqslant 0
\end{aligned} \tag{17}$$

Let $L(\mathbf{x}, \bar{\mathbf{y}}^k) = F(\mathbf{x}, \bar{\mathbf{y}}^k) + \lambda_k^T G(\mathbf{x}, \bar{\mathbf{y}}^k)$, then we can obtain a feasible cut when the primal problem is feasible, which can be added to the master problem as a constraint condition.

$$P^l(\mathbf{y}) = L(\mathbf{x}^\mathbf{k}, \bar{\mathbf{y}}^k) + \nabla_y^T L(\mathbf{x}^\mathbf{k}, \bar{\mathbf{y}}^k)(\mathbf{y} - \bar{\mathbf{y}}^k). \tag{18}$$

If the primal problem $P2.1$ is infeasible with respect to $\bar{\mathbf{y}}^k$, we can relax the primal problem into the following subproblem.

$$\begin{aligned}
P2.2: \quad & \min_{\mathbf{x}} \sum_{j=1}^{4N+3} \alpha_j \\
s.t. \quad & G(\mathbf{x}, \bar{\mathbf{y}}^k) - \alpha \leqslant 0 \\
& \alpha_j \geqslant 0, \forall j
\end{aligned} \tag{19}$$

where α is the introduced relaxation vector. Its dimension is $4N+3$, the same to the size of $G(\mathbf{x},\bar{\mathbf{y}}^k)$. In this case, the infeasible primal problem is transformed to a feasible one. Through solving the subproblem $P2.2$, we can obtain a continuous solution of $\mathbf{x}^k$ and the Lagrange multiplier vector η_k for the constraints.

To accelerate the convergence of the relaxation subproblem, the infeasible cut of the primal problem can be given by the product of η_k and the first order Taylor of approximation for $G(\mathbf{x},\mathbf{y})$ at the point $(\mathbf{x}^k,\bar{\mathbf{y}}^k)$, which is similar to the outer approximation method [34]. The infeasible cut can be obtained by

$$Q^m(\mathbf{x},\mathbf{y}) = \eta_k^T(G(\mathbf{x}^k,\bar{\mathbf{y}}^k)+\nabla_{x,y}^T G(\mathbf{x}^k,\bar{\mathbf{y}}^k)((\mathbf{x},\mathbf{y})-(\mathbf{x}^k,\bar{\mathbf{y}}^k))). \tag{20}$$

Based on the feasible cut and the infeasible cut, the master problem to solve the binary variables can be formulated as

$$\begin{aligned} P2.3: \quad & \min_{\mathbf{y}} z \\ s.t. \quad & P^l(\mathbf{y}) \leqslant z, \forall l \in F \\ & Q^m(\mathbf{x},\mathbf{y}) \leqslant 0, \forall m \in IF \\ & \sum_{i\in B^m} y_i - \sum_{i\in NB^m} y_i \leqslant |B^m| - 1, \forall m \in IF \\ & (B^m = \{i : y_i^m = 1\}, NB^m = \{i : y_i^m = 0\}) \\ & y_i \in \{0,1\}, \forall i \in \{1,2,...2N\} \end{aligned} \tag{21}$$

where F and IF are the sets of the feasible primal problem and the relaxation subproblem, respectively. Herein, $F = \{l | l \leqslant k, \text{problem } P2.1(\bar{\mathbf{y}}^k) \text{ is feasible}\}$ and $IF = \{m | m \leqslant k, \text{problem } P2.1(\bar{\mathbf{y}}^k) \text{ is infeasible}\}$. The third constraint of the master problem is an integer cut [35], [36], which is added to exclude the infeasible binary vector. After the finite K iterations of MGBD, the optimal objective function value of the master problem can be obtained, which is also the lower bound of problem $P2$, namely $LB^K = z^K$.

Lemma 1: If $\mathbf{s}$ and $\mathbf{s}^*$ are the optimal offloading decisions of problem $P2$ and problem $P0$ respectively, $\sum_{i=1}^{N} s_i \geqslant \sum_{i=1}^{N} s_i^*$ holds.

Proof: We define $t_i^M = \frac{d_i}{a_i B r_i} + \frac{e_i}{b_i F} + (1-c_i)\frac{f_i}{R_i^b}$ and $t_i^{M1} = \frac{d_i}{(a_i+\varepsilon_1)B r_i} + \frac{e_i}{(b_i+\varepsilon_1)F} + (1-c_i)\frac{f_i}{R_i^b}$ as the generated edge latency of SMD i for problem $P0$ and $P2$, respectively. The local latency consumption is denoted by $t_i^L = \frac{e_i}{f_i^l}$. In order to obtain the minimum latency consumption $t_i = s_i t_i^M + (1-s_i)t_i^L$, the optimal offloading decision is decided by the difference between t_i^M and t_i^L. We assume that the value of c_i is fixed, because it is not affected by the introduced microscales.

For problem $P0$, if $t_i^M - t_i^L > 0$, the optimal offloading decision should be determined as $s_i^* = 0$. Otherwise, $s_i^* = 1$. However, due to the introduction of microscales ε_1 and ε_2, the edge latency of SMD i in problem $P2$ is sensitive to them. t_i^{M1} should be less than t_i^M in problem $P0$ for the same resource allocated. As a result, if $t_i^{M1} - t_i^L \leqslant 0$ maybe hold for $t_i^M - t_i^L > 0$, then $s_i = 1$ holds. Moreover, $s_i = 1$ and $t_i^{M1} - t_i^L < 0$ should hold when $t_i^M - t_i^L < 0$.

Therefore, we draw a conclusion that the number of tasks offloaded to the MEC sever of problem $P2$ is no less than that of problem $P0$. ■

According to $Lemma$ 1, we design the joint computation offloading, caching and resource allocation algorithm based on the MGBD method in Algorithm 1.

Algorithm 1: Optimal computation offloading, content caching and resource allocation via the MGBD method

Initialization: $\bar{\mathbf{y}}^0$, $UB = +\infty$, $LB = -\infty$, $Ts = +\infty$.
while $|UB - LB|/LB > \varepsilon$ **do**
  Solve the primal problem $P2.1$. If $P2.1(\bar{\mathbf{y}}^k)$ is feasible, the continuous solution $\mathbf{x}^k$ and the Langrange multiplier λ_k can be obtained.
  Caculate the upper bound $UB = F(\mathbf{x}^k,\bar{\mathbf{y}}^k)$ and the feasible cut $P^l(\mathbf{y})$.
  If $P2.1(\bar{\mathbf{y}}^k)$ is infeasible, solve the relaxation subproblem $P2.2$ to obtain $\mathbf{x}^k$ and the Lagrange multiplier η_k.
  Caculate the infeasible cut $Q^m(\mathbf{x},\mathbf{y})$.
  Solve the master problem $P2.3$ to obtain the binary vector $\mathbf{y}$ and the lower bound $LB = z^k$.
  Update k, l, m.
end
The binary vector $\mathbf{y}$ can be obtained through iterations.
for $n = 1 : \sum_{i=1}^{N} y_i$ **do**
  Solve problem $P1$ based on the binary vector $\mathbf{y}$.
  If $P1$ is feasible, the continuous vector $\mathbf{x}$ and the optimal function value T can be obtained.
  If $P1$ is infeasible, $T = T_s$.
  if $T <= T_s$ **then**
    $T_s = T$, $\mathbf{x}_s = \mathbf{x}$, $\mathbf{y}_s = \mathbf{y}$.
    Sort all the partial derivations $\nabla_y^T L(\mathbf{x}_s,\mathbf{y}_s)$ of SMDs with offloading tasks from large to small.
    Set the offloading decision with the largest partial derivations as 0 and modify the caching decision according to constraints of $C5$ and $C6$.
    Update the value of $\mathbf{y}$.
  end
end
Output the optimal solutions of the problem $P1$ $\mathbf{x}_s$, $\mathbf{y}_s$, T_s.

Algorithm 1 is mainly composed of two main procedures, i.e., decision making and checking. The primary computation offloading and caching decisions are determined by the MGBD method at first. The MGBD method terminates in a finite number of iterations according to the convex set of continuous variables and the convex objective function together with constraints for the fixed binary variables [37]. The stop criterion is that the gap of lower bound and upper bound is less than a small value ε [38]. Then, we check each offloading decision and caching decision based on problem $P1$. If removing the offloading task can reduce the total latency, the corresponding task will be transformed to local execution. The check operation terminates until the total latency stops decreasing. After check operation is finished, the optimal decision and resource allocation strategy can be determined simultaneously.

C. Summary of MBB and MGBD

In problem $P0$, if all integer variables equal to 0, namely the extreme case that all tasks are computed locally happens, then the proposed algorithms are infeasible. However, in fact, the computation and storage capabilities of the MEC server are always higher than that of SMDs, and at least one task can be executed on the server. When problem $P0$ is converted as problem $P1$, the problem is convex. MBB and MGBD can be used to solve the problem. Both of them adopt the similar idea of separating the binary and continuous variables to solve the problem. MBB adopts the depth-first search strategy to search the binary solutions at first, then the continuous solutions can be determined by the convex optimization algorithms. The main difference is that MGBD can find the optimal binary and continuous solutions through several iterations. The computation complexity of MBB depends mainly on the number of binary variables, while that of MGBD is affected by the number of iterations.

The complexity of the proposed algorithms is detailed as follows. To determine the value of one integer variable, the MBB method needs to solve a nonlinear problem. Traversing a path of the search tree, $2N$ nonlinear problems should be solved. There are 3^N paths in the whole asymmetric tree, which can decrease $2^{2N} - 3^N$ paths compared with the completed tree. In the worst case, $2N3^N$ nonlinear problems should be solved for the problem and the computation complexity of the MBB method can reach $O(3^N)$. Hence, the MBB method still has an exponential computation complexity. Assume that the MGBD method stops through K iterations, K nonlinear primal problems and K mixed integer linear master problems can be solved. If the number of the check operation for the decision variables is N_c, then N_c nonlinear convex problems can be solved, satisfying $N_c \leqslant \sum_{i=1}^{N} y_i$. Therefore, $(K+N_c)$ NLP problems and K MILP problems can be solved to obtain the optimal decision and resource allocation strategy, which has a polynomial complexity.

V. Simulation Results and Discussions

In this section, we consider a centralized MEC network covered with a $200m \times 200m$ area, where the MeNB is located at the center and SMDs are randomly scattered over the region. Each SMD has a computation task to be executed. The input data size of the task is randomly distributed within [100, 1000] KB, and the corresponding number of the required CPU cycles is distributed within [0.2, 1] $Gcycles$. The size of requested content is 0.1 Mb and the remaining caching capability in the MEC server is 0.6 Mb. The transmission power of SMDs is set to 100 mW when offloading the tasks to the MEC server. The path loss model from SMD to MeNB follows the 3GPP specification [39]. The total types of the Internet contents are 50, and the average transmission rate between the Internet and the MEC server is 100 $Mbps$. The tolerance latency of SMDs is randomly distributed between 0.2 s and 1 s. The main communication and computation parameters employed in the simulations are illustrated in Table II.

TABLE II: Simulation parameters.

Parameters	Value
Total bandwidth	16 MHz
Transmission power of SMDs	100 mW
Path loss model from SMD to the MeNB	$128.1 + 37.5\log_{10}(R)$ dB
Thermal Noise density	-174 dBm/Hz
Local computation capacity	1.25 GHz
Edge computation capacity	50 GHz

Furthermore, simulation experiments are developed for the proposed algorithms in comparison with the centralized algorithm [19] and several baselines. 'All local' represents that all tasks are locally executed. 'All offloading' indicates that all tasks are offloaded to the MEC server, only content caching and resource allocation are solved. 'Equal spectrum resource' and 'Equal computation resource' show that the spectrum and computation resources are equally allocated to all tasks. System performance and resource allocation are evaluated in this section.

A. System performance

In Fig. 3, we investigate the system latency performance with respect to the number of SMDs in four schemes. The total latency increases as the number of SMDs enhances. When all tasks of SMDs are executed locally (i.e., All local), the latency consumption is always higher than the proposed schemes. The reason is that SMDs have the limited computation capability. The latency consumption obtained by all offloading, centralized algorithm, MBB and MGBD is the same for $N = 4$, which means the MEC server has enough computation capacity to handle all tasks. The difference among them becomes greater with the increase of the number of SMDs. The latency consumption for all offloading increases dramatically and even surpasses that for all local when $N \geqslant 12$. It can be explained that fewer resources are allocated to each offloading task for the limited total communication and computation resources. On the other hand, the minimum latency consumption is achieved by the proposed MBB algorithm. The latency gap between MBB and MGBD is small and within 0.03 s, while the latency difference between the centralized algorithm and MBB is within 0.2 s. It can be observed that both of the proposed MBB and MGBD algorithms outperform other schemes. In addition, we also adopt the proposed schemes without content caching to reveal the effect of content caching in Fig. 3. Note that our model is converted to the joint optimization problem of computation offloading and resource allocation when there is no content caching. Then, only offloading decisions remain in the integer variables, and we can establish a symmetrical search tree and adopt the classical branch and bound (BB) method to solve the problem, i.e., BB (no-caching). The model and approach are similar to [6] except for the objective function. As shown in Fig. 3, the latency gap between MBB and BB (no-caching) (or MGBD and MGBD (no-caching)) is about 0.3 s. Due to the fact that caching contents from the Internet can alleviate the backhaul bandwidth, the proposed algorithms with content caching can decrease more latency than that without content caching.

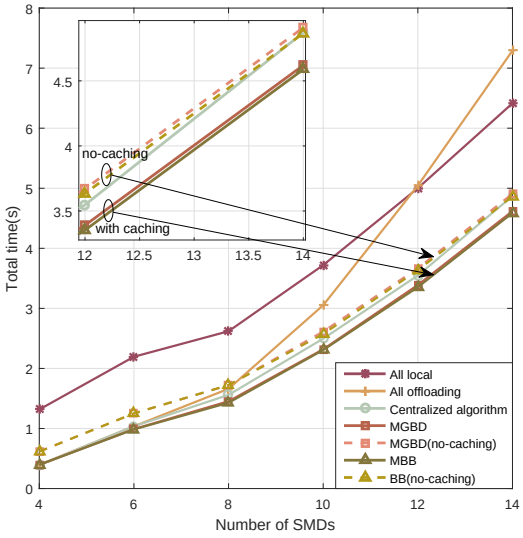

Fig. 3: Latency consumption versus the number of SMDs.

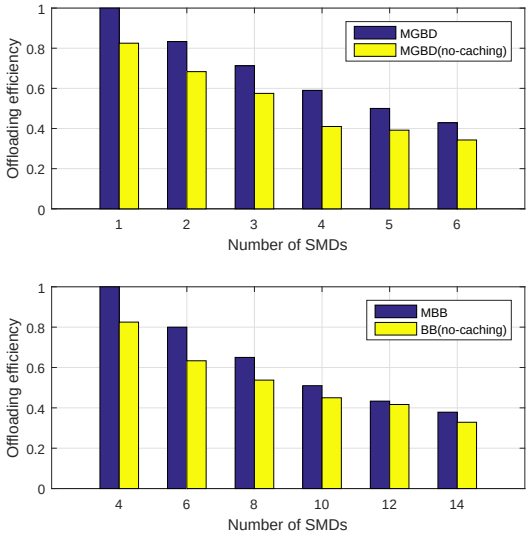

Fig. 4: Offloading efficiency versus the number of SMDs.

We further discuss the offloading efficiency in the proposed algorithms with and without content caching, as shown in Fig. 4. The offloading efficiency is defined as the ratio of the number of SMDs with offloading tasks to the total number of SMDs. From Fig. 4, we can find that the offloading efficiency obtained by the proposed algorithms with content caching is higher than that obtained by the proposed algorithms without content caching. The reason is that content caching can save the backhaul delay, which allows more SMDs to offload their tasks for edge computing. Besides, it can also be seen that the offloading efficiency obtained by MGBD is higher than that obtained by MBB in most cases. Associating the fact that the latency of MGBD is larger than that of MBB in Fig. 3, it indicates that the high offloading efficiency does not lower latency consumption, which is attributed to the limited communication and computation resources. As fewer resources are allocated to each SMD, the total latency consumption grows. Hence, it is of great importance to find the optimal offloading decision and caching decision.

To demonstrate the complexity analysis of the proposed algorithms, we show the running time of MBB and MGBD in Fig. 5. From Fig. 5, the running time of the proposed algorithms with content caching is always more than that of the proposed algorithms without content caching, which is attributed to the number of the unknown variables. The proposed algorithms with content caching have more binary variables, resulting in more computation processes. As the number of SMDs increases, the running time of MBB and BB (no-caching) increases dramatically and shows an exponential growth. On the other hand, the running time of MGBD and MGBD (no-caching) shows a slow rising tendency. Besides, we can observe that the gap between MGBD and MBB becomes wider with the increase of the number of SMDs.

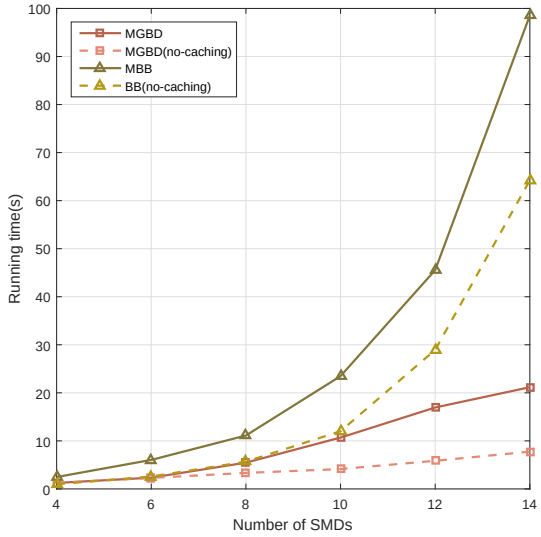

Fig. 5: Running time of MBB and MGBD.

B. Resource allocation

In our model, spectrum and computation resources allocation are comprehensively considered. To investigate the effect of resource allocation on the latency performance, we compare the proposed algorithms with the centralized algorithm as well as equal resource allocation. Fig. 6 illustrates the latency consumption with respect to the increasing spectrum resource, where the number of SMDs is selected as 10 and the computation capability of the MEC server is set to 50 GHz. In Fig. 6, the proposed MBB algorithm can still achieve the optimal latency performance. The latency gap between MGBD and MBB is within 0.03s, smaller than that between the centralized algorithm and MBB. The latency consumption of equal spectrum resource is more than the other three schemes. Due to the underlying manifold of tasks, equal spectrum

resource allocation cannot provide more resources for the large computation tasks but offer overmuch resources for the small computation tasks, leading to additional latency and resource waste. In addition, the latency consumption decreases with the increase of spectrum resource and the latency gap among four schemes has a decreasing tendency in total. The reason is that more spectrum resources can enable the decrease of the transmission time, which plays an important role in the total latency consumption. In an extreme situation, all SMDs can offload their tasks to the MEC server when the spectrum resource is large enough.

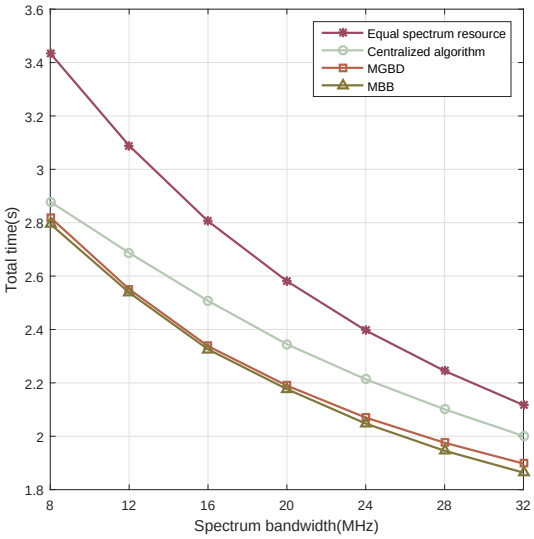

Fig. 6: Latency consumption versus spectrum resource ($N = 10$, $F = 50GHz$).

Fig. 7 shows the latency consumption with respect to the increasing computation resource of the MEC server. The latency consumption decreases with the increase of computation resource. The reason is that more computation resource can save more task execution time. However, the descent speed of latency consumption with the increase of computation capability is much less than that with the increase of spectrum resource. In particular, the latency consumption just decreases by 0.15 s with the increase of computation capability from 42 GHz to 66 GHz, far less than 1 s obtained by the increase of the spectrum resource through the observation on the MBB and MGBD curves. The tendency can be explained by the negligible task execution time in the MEC server. Due to the fact that the computation capability of the MEC server is more than that of SMDs, the task execution time of edge computing is less than that of local computing. Hence, the proportion of the task execution time to the total latency consumption is very small compared with the transmission time. Besides, similar to Fig. 6, the proposed methods are superior to the other two schemes in Fig. 7.

In Fig. 8, the allocation of a group of communication and computation resources is detailed. We choose a MEC network scenario with 10 SMDs, where $B = 16\ MHz$ and $F = 50\ GHz$. In this case, we list resource allocation result of SMDs with offloading tasks in the order from small to large. From

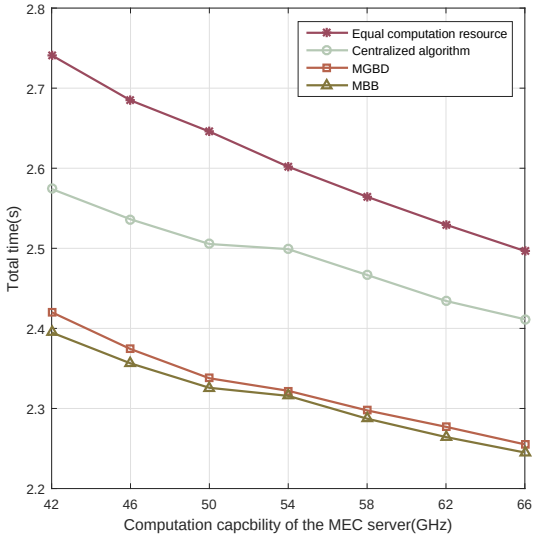

Fig. 7: Latency consumption versus computation capability of the MEC server ($N = 10$, $B = 16MHz$).

Fig. 8, we can see that the sum of the radio of resources allocated to all offloading tasks is approximately equal to 1 for the proposed algorithm and the centralized algorithm, namely almost all of communication and computation resources can be utilized for these algorithms. However, not all SMDs can offload their tasks to the MEC server for the equal resource allocation scheme, leading to resource waste. This situation also explains why equal resource allocation scheme always obtains the largest latency consumption in Fig. 6 and Fig. 7. On the other hand, due to more SMDs with offloading tasks for the centralized algorithm, fewer resources are allocated to each SMD compared with MBB and MGBD. Although the number of SMDs with offloading tasks for MBB and MGBD is the same, the allocated resources are still different. It can be explained that SMDs with offloading tasks for MBB is not consistent with that for MGBD. In addition, the gap of the allocated communication resource between MBB and MGBD is generally greater than that of the allocated computation resource for one task, which further indicates that the difference of network latency between MBB and MGBD is mainly affected by the transmission time.

VI. Conclusion

In this paper, the MEC network with computation capability and content caching is investigated to improve QoE of SMDs with latency-sensitive computation tasks. We jointly formulate computation offloading, content caching, spectrum and computation resource allocation as an optimization problem. The objective of the formulated problem is to minimize the total latency consumption of all computation tasks. To solve the formulated MINLP problem, we adopt the MBB method to obtain a set of accurate solutions. Moreover, the MGBD is applied to obtain the suboptimal solutions with polynomial computation complexity. Simulation results demonstrate that the proposed algorithms outperform the baseline methods

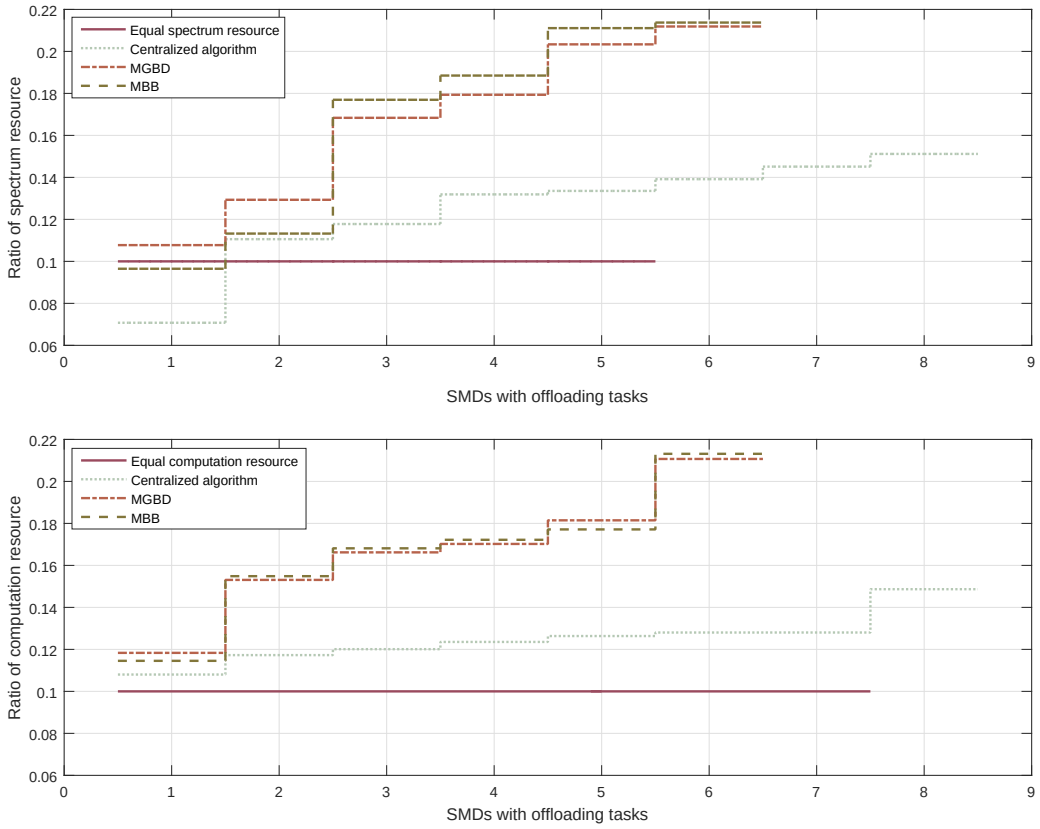

Fig. 8: Ratio of resource allocated versus SMDs with offloading tasks.

under various system parameters. In the future, the integration of computation offloading, content caching and resource allocation in the heterogeneous MEC networks deserves to be explored.

VII. Acknowledgement

This work was partially supported by the Shenzhen-Hongkong Innovative Project (SGLH20161212140718841), Shenzhen Engineering Laboratory for 3D Content Generating Technologies (NO.[2017]476), the key research plan of Hunan province under Grant 2016JC2021, Guangdong Technology Project (2016B010108010, 2016B010125003, 2017B010110007), National Basic Research Program of China (973 Program) (No.2014CB744600), National Nature Science Foundation of China (Nos.61601482, 61403365, 61402458, 61502075, 61632014, 61772508), Program of International S&T Cooperation of MOST (No.2013DFA11140).

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.

[3] J. Pan and J. Mcelhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[5] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[6] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[8] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[9] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444–1462, 2014.

[10] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1382–1393, 2017.

[11] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "A survey of web caching

and prefetching," *International Journal of Advances in Soft Computing & Its Application*, vol. 3, no. 1, 2011.
[12] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.
[13] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5g ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. 974, 2016.
[14] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, no. 99, pp. 6757–6779, 2017.
[15] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Wireless On-Demand Network Systems and Services*, pp. 165–172, 2017.
[16] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultra-dense iot networks," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2018.
[17] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.
[18] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.
[19] Z. Tan, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Virtual resource allocation for heterogeneous services in full duplex-enabled scns with mobile edge computing and caching," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.
[20] W. Fan, Y. Liu, B. Tang, F. Wu, and H. Zhang, "Exploiting joint computation offloading and data caching to enhance mobile terminal performance," in *GLOBECOM Workshops*, pp. 1–6, 2017.
[21] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2018.
[22] N. Zhao, X. Liu, F. R. Yu, M. Li, and V. C. M. Leung, "Communications, caching, and computing oriented small cell networks with interference alignment," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 29–35, 2016.
[23] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010.
[24] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8744–8751, 2016.
[25] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," *Proc of IEEE Infocom Mar*, vol. 1, pp. 126 – 134, 1999.
[26] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.
[27] X. Wang, M. Chen, T. Taleb, and A. Ksentini, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.
[28] Y. Pochet and L. A. Wolsey, "Production planning by mixed integer programming," *Springer*, vol. 149, no. 2, pp. 163–175, 2006.
[29] Boyd, Vandenberghe, and Faybusovich, "Convex optimization," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1859–1859, 2006.
[30] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.
[31] R. Hemmecke, M. Kppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," *Computers & Mathematics with Applications*, vol. 1, no. 2, pp. 215–222, 2012.
[32] M. J. Bagajewicz and V. Manousiouthakis, "On the generalized benders decomposition," *Computers & Chemical Engineering*, vol. 15, no. 10, p. 691700, 1991.
[33] S. Rahimi, T. Niknam, and F. Fallahi, "A new approach based on benders decomposition for unit commitment problem," *World Applied Sciences Journal*, no. 12, 2013.
[34] R. Fletcher and S. Leyffer, "Solving mixed integer nonlinear programs by outer approximation," *Mathematical Programming*, vol. 66, no. 1-3, pp. 327–349, 1994.
[35] E. Balas and R. Jeroslow, "Canonical cuts on the unit hypercube," *Siam Journal on Applied Mathematics*, vol. 23, no. 1, pp. 61–69, 1972.
[36] P. Kesavan, R. J. Allgor, E. P. Gatzke, and P. I. Barton, "Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs," *Mathematical Programming*, vol. 100, no. 3, pp. 517–535, 2004.
[37] C. A. Floudas, "Nonlinear and mixed-integer optimization [electronic resource] : fundamentals and applications /," *Handbook of Turbulence*, vol. 11, no. 00, p. 822823, 1995.
[38] E. Muoz and M. Stolpe, "Generalized benders decomposition for topology optimization problems," *Journal of Global Optimization*, vol. 51, no. 1, pp. 149–183, 2011.
[39] 3GPP, "Evolved universal terrestrial radio access (E-UTRA): Radio frequency(RF) system scenarios," *TR 36.942 V11. 0. 0*, 2012.