

Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching

Jiao Zhang¹, Xiping Hu¹, Zhaolong Ning¹, *Member, IEEE*, Edith C.-H. Ngai², *Senior Member, IEEE*,
Li Zhou¹, Jibo Wei, Jun Cheng¹, Bin Hu¹, *Senior Member, IEEE*,
and Victor C. M. Leung³, *Fellow, IEEE*

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 mixed 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—Content caching, Internet of Things (IoT), mobile edge computing (MEC), resource allocation.

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 [2]. 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 [3]. 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 [4], [5].

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 [6], [7]. 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 [8]–[11], 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

Manuscript received July 19, 2018; accepted October 6, 2018. Date of publication October 15, 2018; date of current version June 19, 2019. This work was supported in part by the Shenzhen-Hongkong Innovative Project under Grant SGLH20161212140718841, in part by the Shenzhen Engineering Laboratory for 3-D Content Generating Technologies under Grant [2017]476, in part by the Key Research Plan of Hunan Province under Grant 2016JC2021 and Grant 2016JC2022, in part by the Guangdong Technology Project under Grant 2016B010108010, Grant 2016B010125003, and Grant 2017B010110007, in part by the National Basic Research Program of China (973 Program) under Grant 2014CB744600, in part by the National Nature Science Foundation of China under Grant 61601482, Grant 61403365, Grant 61402458, Grant 61502075, Grant 61632014, and Grant 61772508, and in part by the Program of International S&T Cooperation of MOST under Grant 2013DFA11140. (Corresponding authors: Xiping Hu; Zhaolong Ning; Li Zhou; Jun Cheng; Bin Hu.)

J. Zhang is with the College of Electronic Science, National University of Defense Technology, Changsha 410073, China, and also with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: zhangjiao16@nudt.edu.cn).

X. Hu and J. Cheng are with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: xp.hu@siat.ac.cn; jun.cheng@siat.ac.cn).

Z. Ning is with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: zhaolongning@dlut.edu.cn).

E. C.-H. Ngai is with the Department of Information Technology, Uppsala University, 75105 Uppsala, Sweden (e-mail: edith.ngai@it.uu.se).

L. Zhou and J. Wei are with the College of Electronic Science, National University of Defense Technology, Changsha 410073, China (e-mail: zhouli2035@nudt.edu.cn; wjbhw@nudt.edu.cn).

B. Hu is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 410073, China (e-mail: bh@lzu.edu.cn).

V. C. M. Leung is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T1Z4, Canada (e-mail: vleung@ece.ubc.ca).

Digital Object Identifier 10.1109/IIOT.2018.2875917

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 [12], [13]. 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 [14], [15]. What is more, the integration of computation offloading and content caching has been attracting more attentions with the emergence of MEC networks [16]–[18].

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.

- 1) We jointly consider the computation offloading, content caching and resource allocation to formulate an optimization problem to improve the quality of user experience for SMDs with latency-sensitive tasks.
- 2) Based on the branch and bound (BB) method and problem constraints, we modify the deep search tree and adopt the improved BB method to solve the optimization problem.
- 3) We decompose the original nonconvex problem and apply the modified generalized benders decomposition (GBD) 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 BB method (MBB) together with the GBD are presented. Simulation results are discussed in Section V. The 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. Zhao *et al.* [9] considered an energy-saving offloading framework by jointly optimizing radio and computation resources in MEC networks. Guo *et al.* [19] studied the problem of mobile edge computation offloading in ultradense IoT networks, so as to minimize the overall computation overhead of all tasks while satisfying the channel constraint. In [20], 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 [13] to minimize the average downloading latency under the constraints of storage and bandwidth capabilities. In [18], 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 [21], 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 [22]. However, both [21] and [22] 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 [23]. However, there is no consideration about communication resource allocation. Chen *et al.* [24] 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 [25].

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, where the spectrum resource allocated to each SMD cannot be reused. The set of SMDs overlaid by MeNB is denoted by $\mathcal{N} = \{1, 2, \dots, N\}$. 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

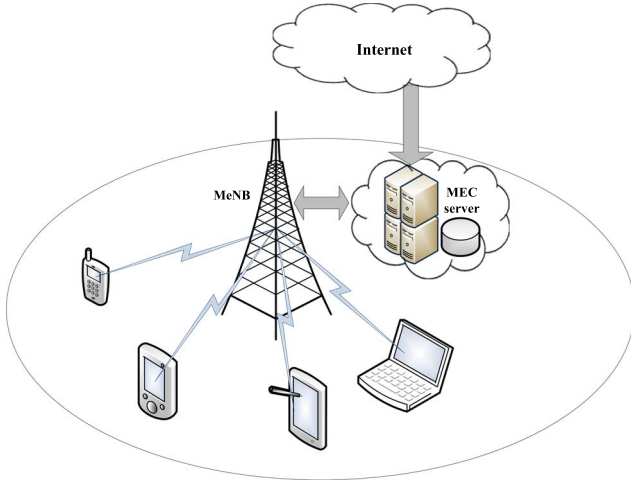


Fig. 1. Illustrative network model.

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 j th content requested by SMD i
R_i^b	The alleviated backhaul bandwidth

been stored, it can be reused. For each SMD, its task can be either computed locally or offloaded to the MEC server for execution.

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.

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

$$r_i = \log_2 \left(1 + \frac{p_i h_i}{\sigma^2} \right) \quad (1)$$

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 [26].

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

$$R_i^u = a_i B r_i. \quad (2)$$

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

$$t_i^u = \frac{d_i}{R_i^u}. \quad (3)$$

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

$$t_i^l = \frac{e_i}{f_i^l}. \quad (4)$$

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

$$t_i^e = \frac{e_i}{b_i F}. \quad (5)$$

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 [27] and [28], we assume that the popularity of the requested contents is modeled as Zipf distribution. Thus, the popularity of the j th popular content requested by SMD i can be given by

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

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 [21], [29].

According to [21] and [30], 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 j th popular content requested by SMD i can be obtained by

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

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

$$t_i^c = \frac{f_i}{R_i^b}. \quad (8)$$

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

$$t_i^M = t_i^u + t_i^e + (1 - c_i)t_i^c \quad (9)$$

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 [10] and [21], 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

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

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:

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

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 $e_i/f_i^l \leq T_i^{\max}$ is always satisfied, so as to guarantee that problem $P0$ has an optimal solution [9].

IV. PROBLEM TRANSFORMATION AND SOLUTION

In this section, we first present some problem observation and reformulation. Then, the 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 (MINLP) [31], 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 $P1$

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

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 [32].

A. Modified Branch and Bound Method

According to the special constitution of problem $P1$, we can adopt the BB method [33], [34] for the original problem $P0$, which is sophisticated for the MINLP problem. The idea of the BB 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

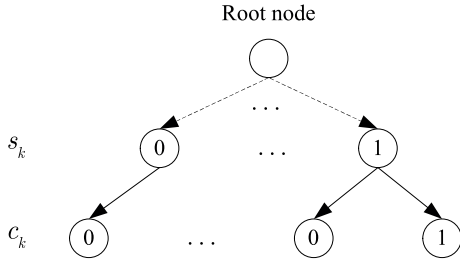


Fig. 2. Search tree of the MBB method.

offloading decision and caching decision (C5) in problem $P0$, we establish an asymmetric search tree, as shown in Fig. 2.

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 l th 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 child 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 [9] and [33].

B. Modified Generalized Benders Decomposition

In order to reduce the computation complexity, the GBD method [35], [36] 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 $P2$

$$\begin{aligned}
 P2 : \quad & \min_{\mathbf{s}, \mathbf{c}, \mathbf{u}, \mathbf{v}} \sum_{i=1}^N \left\{ s_i \left(\frac{d_i}{u_i B r_i} + \frac{e_i}{v_i F} + (1 - c_i) \frac{f_i}{R_i^b} \right) + (1 - s_i) \frac{e_i}{f_i^l} \right\} \\
 \text{s.t.} \quad & C1 : \varepsilon_1 \leq u_i \leq s_i + \varepsilon_1 \quad \forall i \\
 & C2 : \sum_{i=1}^N u_i \leq 1 + N\varepsilon_1 \\
 & C3 : \varepsilon_2 \leq v_i \leq s_i + \varepsilon_2 \quad \forall i \\
 & C4 : \sum_{i=1}^N v_i \leq 1 + N\varepsilon_2
 \end{aligned}$$

$$\begin{aligned}
 C7 : \quad & s_i \left(\frac{d_i}{u_i B r_i} + \frac{e_i}{v_i F} + (1 - c_i) \frac{f_i}{R_i^b} \right) + (1 - s_i) \frac{e_i}{f_i^l} \\
 & \leq T_i^{\max} \quad \forall i \\
 & C5, C6, C8, C9.
 \end{aligned} \tag{13}$$

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

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

For a fixed realization of the binary variables, 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

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

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

problem $P2.1$

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

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

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

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

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

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:

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

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 [37]. The infeasible cut can be obtained by

$$\begin{aligned} Q^m(\mathbf{x}, \mathbf{y}) = & \eta_k^T \left(G(\mathbf{x}^k, \bar{\mathbf{y}}^k) + \nabla_{\mathbf{x}, \mathbf{y}}^T G(\mathbf{x}^k, \bar{\mathbf{y}}^k) \right. \\ & \left. \times \left((\mathbf{x}, \mathbf{y}) - (\mathbf{x}^k, \bar{\mathbf{y}}^k) \right) \right). \end{aligned} \quad (20)$$

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

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

where F and IF are the sets of the feasible primal problem and the relaxation subproblem, respectively. Herein, $F = \{l | l \leq k, \text{problem } P2.1(\bar{\mathbf{y}}^k) \text{ is feasible}\}$ and $\text{IF} = \{m | m \leq k, \text{problem } P2.1(\bar{\mathbf{y}}^k) \text{ is infeasible}\}$. The third constraint of the master problem is an integer cut [38], [39], 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, $\sum_{i=1}^N s_i \geq \sum_{i=1}^N s_i^*$ holds.

Proof: We define $t_i^M = [d_i / (a_i B r_i)] + (e_i / b_i F) + (1 - c_i)(f_i / R_i^b)$ and $t_i^{M1} = [d_i / ((a_i + \varepsilon_1) B r_i)] + [e_i / ((b_i + \varepsilon_1) F)] + (1 - c_i)(f_i / R_i^b)$ as the generated edge latency of SMD i for problem $P0$ and $P2$, respectively. The local latency consumption is denoted by $t_i^L = (e_i / f_i^l)$. 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 microscale.

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 microscale 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 $t_i^{M1} - t_i^L \leq 0$ 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 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 [40]. The stop criterion is that the gap of lower bound and UB is less than a small value ε [41]. 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

Algorithm 1: Optimal Computation Offloading, Content Caching, and Resource Allocation via the MGBD Method

```

1 Initialization:  $\bar{\mathbf{y}}^0$ ,  $UB = +\infty$ ,  $LB = -\infty$ ,  $T_s = +\infty$ .
2 while  $|UB - LB|/LB > \varepsilon$  do
3   Solve the primal problem P2.1. If P2.1( $\bar{\mathbf{y}}^k$ ) is
   feasible, the continuous solution  $\mathbf{x}^k$  and the
   Lagrange multiplier  $\lambda_k$  can be obtained.
4   Calculate the upper bound  $UB = F(\mathbf{x}^k, \bar{\mathbf{y}}^k)$  and the
   feasible cut  $P^l(\mathbf{y})$ .
5   If P2.1( $\bar{\mathbf{y}}^k$ ) is infeasible, solve the relaxation
   subproblem P2.2 to obtain  $\mathbf{x}^k$  and the Lagrange
   multiplier  $\eta_k$ .
6   Calculate the infeasible cut  $Q^m(\mathbf{x}, \mathbf{y})$ .
7   Solve the master problem P2.3 to obtain the binary
   vector  $\mathbf{y}$  and the lower bound  $LB = z^k$ .
8   Update  $k$ ,  $l$ ,  $m$ .
9 end
10 The binary vector  $\mathbf{y}$  can be obtained through iterations.
11 for  $n = 1 : \sum_{i=1}^N y_i$  do
12   Solve problem P1 based on the binary vector  $\mathbf{y}$ .
13   If P1 is feasible, the continuous vector  $\mathbf{x}$  and the
   optimal function value  $T$  can be obtained.
14   If P1 is infeasible,  $T = T_s$ .
15   if  $T \leq T_s$  then
16      $T_s = T$ ,  $\mathbf{x}_s = \mathbf{x}$ ,  $\mathbf{y}_s = \mathbf{y}$ .
17     Sort all the partial derivations  $\nabla_{\mathbf{y}}^T L(\mathbf{x}_s, \mathbf{y}_s)$  of
   SMDs with offloading tasks from large to small.
18     Set the offloading decision with the largest partial
   derivations as 0 and modify the caching decision
   according to constraints of C5 and C6.
19     Update the value of  $\mathbf{y}$ .
20   end
21 end
22 Output the optimal solutions of the problem P1  $\mathbf{x}_s$ ,  $\mathbf{y}_s$ ,  $T_s$ .
```

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

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

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 $N_c \leq \sum_{i=1}^N y_i$. 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 DISCUSSION

In this section, we consider a centralized MEC network covered with a 200 m $\times$ 200 m 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 [42]. The total types of the Internet contents are 50, and the average transmission rate between the Internet and the MEC server is 100 Mb/s. 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.

Furthermore, simulation experiments are developed for the proposed algorithms in comparison with the centralized algorithm [22] 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.

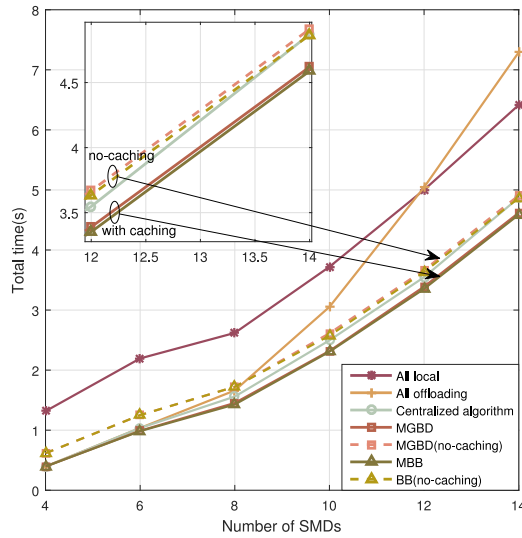


Fig. 3. Latency consumption versus the number of SMDs.

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 $N \geq 12$. 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 BB method to solve the problem, i.e., BB (no-caching). The model and approach are similar to [9] 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.

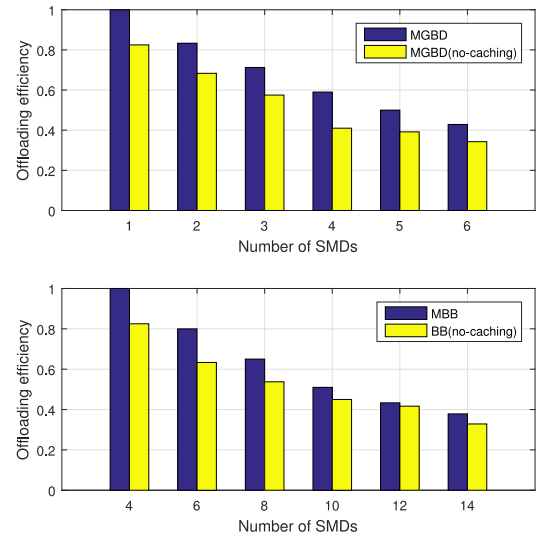


Fig. 4. Offloading efficiency versus the number of SMDs.

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.

B. Resource Allocation

In our model, spectrum and computation resources allocation are comprehensively considered. To investigate the effect

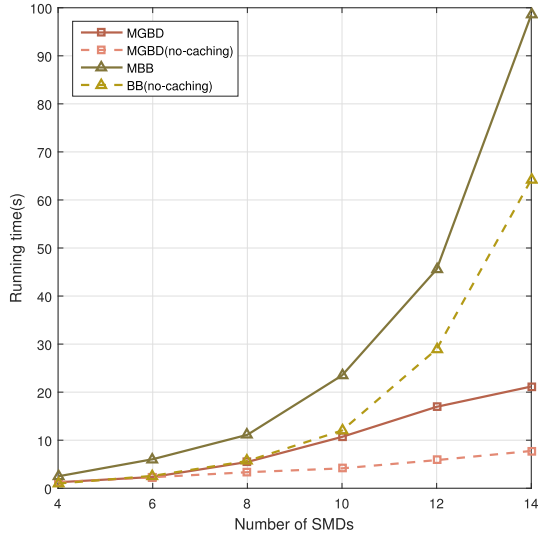
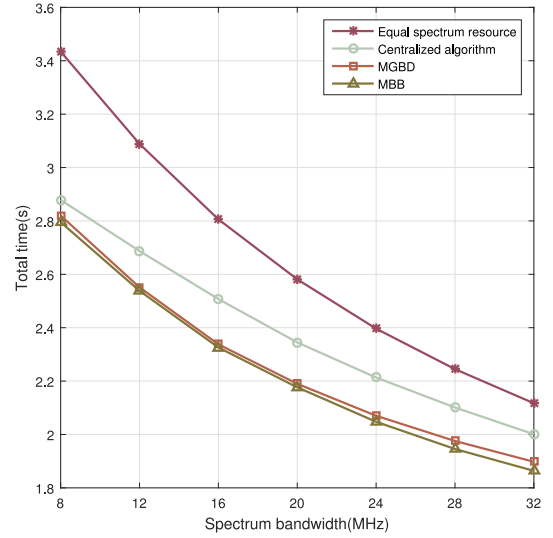
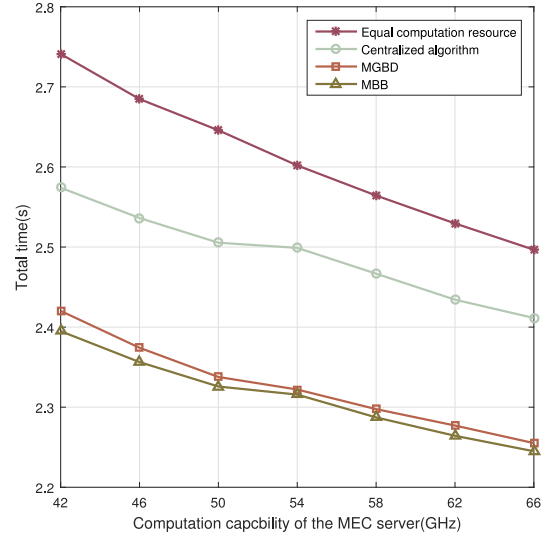


Fig. 5. Running time of MBB and MGBD.

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 0.03 s, 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. 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 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

Fig. 6. Latency consumption versus spectrum resource ($N = 10$, $F = 50$ GHz).Fig. 7. Latency consumption versus computation capability of the MEC server ($N = 10$, $B = 16$ MHz).

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 an 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. 8, we can see that the sum of the ratio 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

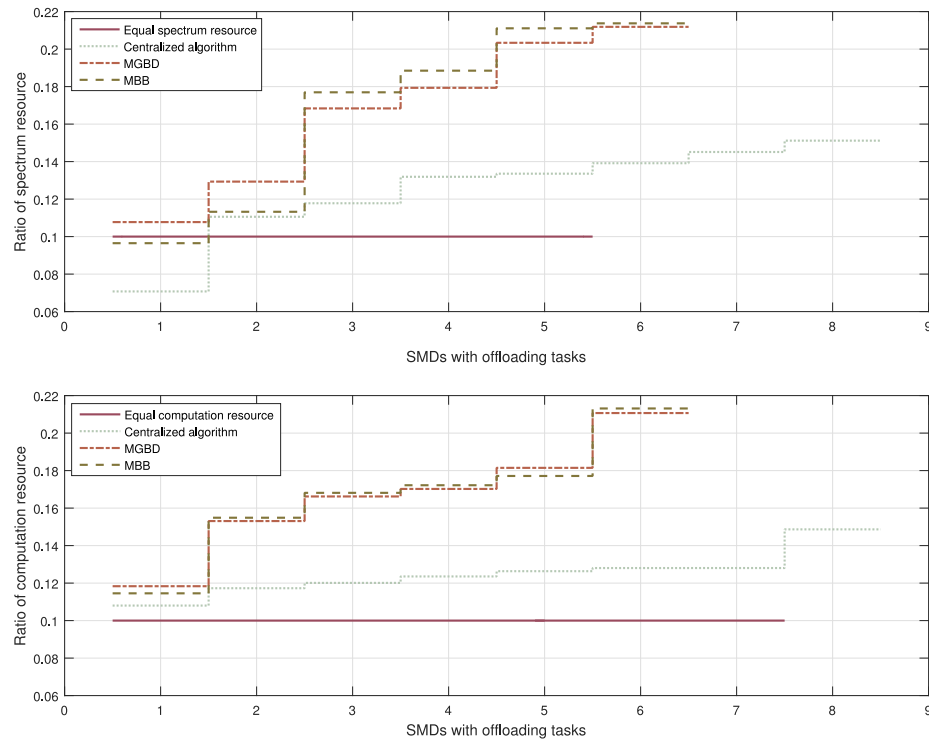


Fig. 8. Ratio of resource allocated versus SMDs with offloading tasks.

explains why equal resource allocation scheme always obtains the largest latency consumption in Figs. 6 and 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 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.

REFERENCES

- [1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [2] Y. Zhang *et al.*, "Cognitive machine-to-machine communications: Visions and potentials for the smart grid," *IEEE Netw.*, vol. 26, no. 3, pp. 6–13, May/Jun. 2012.
- [3] 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 Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1294–1313, 3rd Quart., 2013.
- [4] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 16–55, May 2017.
- [5] Y. Zhang *et al.*, "Home M2M networks: Architectures, standards, and QoS improvement," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 44–52, Apr. 2011.
- [6] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, to be published, doi: [10.1109/IIOT.2018.2868616](https://doi.org/10.1109/IIOT.2018.2868616).
- [7] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.
- [8] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
- [9] 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. 5, pp. 11255–11268, 2017.
- [10] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
- [11] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comput.*, to be published, doi: [10.1109/TCC.2016.2560808](https://doi.org/10.1109/TCC.2016.2560808).
- [12] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.
- [13] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, May 2017.

- [14] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.
- [15] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar, "Dependable demand response management in the smart grid: A Stackelberg game approach," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 120–132, Mar. 2013.
- [16] M. Chen *et al.*, "Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. E974, 2016.
- [17] S. Wang *et al.*, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
- [18] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. Wireless Demand Netw. Syst. Services*, 2017, pp. 165–172.
- [19] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultra-dense IoT networks," *IEEE Internet Things J.*, to be published, doi: [10.1109/JIOT.2018.2838584](https://doi.org/10.1109/JIOT.2018.2838584).
- [20] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
- [21] 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 Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.
- [22] 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 Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1794–1808, Feb. 2018.
- [23] W. Fan, Y. Liu, B. Tang, F. Wu, and H. Zhang, "Exploiting joint computation offloading and data caching to enhance mobile terminal performance," in *Proc. GLOBECOM Workshops*, 2017, pp. 1–6.
- [24] Q. Chen *et al.*, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 274–287, Feb. 2018.
- [25] 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 Commun. Mag.*, vol. 54, no. 9, pp. 29–35, Sep. 2016.
- [26] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.
- [27] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8744–8751, Oct. 2016.
- [28] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 1999, pp. 126–134.
- [29] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
- [30] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.
- [31] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, vol. 149. New York, NY, USA: Springer, 2006, pp. 163–175.
- [32] S. Boyd, L. Vandenberghe, and L. Faybusovich, "Convex optimization," *IEEE Trans. Autom. Control*, vol. 51, no. 11, p. 1859, Nov. 2006.
- [33] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.
- [34] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," *Comput. Math. Appl.*, vol. 1, no. 2, pp. 215–222, 2012.
- [35] M. J. Bagajewicz and V. Manousiouthakis, "On the generalized benders decomposition," *Comput. Chem. Eng.*, vol. 15, no. 10, p. 691–700, 1991.
- [36] S. Rahimi, T. Niknam, and F. Fallahi, "A new approach based on benders decomposition for unit commitment problem," *World Appl. Sci. J.*, vol. 6, no. 12, pp. 1665–1672, 2013.
- [37] R. Fletcher and S. Leyffer, "Solving mixed integer nonlinear programs by outer approximation," *Math. Program.*, vol. 66, nos. 1–3, pp. 327–349, 1994.
- [38] E. Balas and R. Jeroslow, "Canonical cuts on the unit hypercube," *SIAM J. Appl. Math.*, vol. 23, no. 1, pp. 61–69, 1972.
- [39] P. Kesavan, R. J. Allgor, E. P. Gatzke, and P. I. Barton, "Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs," *Math. Program.*, vol. 100, no. 3, pp. 517–535, 2004.
- [40] C. A. Floudas, "Nonlinear and mixed-integer optimization [electronic resource]: Fundamentals and applications," *Handbook of Turbulence*, vol. 11. New York, NY, USA: Oxford Univ. Press, 1995, pp. 822–823.
- [41] E. Muñoz and M. Stolpe, "Generalized benders' decomposition for topology optimization problems," *J. Glob. Optim.*, vol. 51, no. 1, pp. 149–183, 2011.
- [42] "Evolved universal terrestrial radio access (E-UTRA): Radio frequency (RF) system scenarios, V11. 0. 0" 3GPP, Sophia Antipolis, France, Rep. TR 36.942, 2012.



Jiao Zhang received the B.S. degree from Xiangtan University, Xiangtan, China, in 2013, and the M.S. degree Xidian University, Xi'an, China, in 2016. She is currently pursuing the Ph.D. degree at the College of Electronic Science, National University of Defense Technology, Changsha, China.

Her current research interests include mobile edge computing and resource allocation in heterogeneous networks.



Xiping Hu received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada.

He is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He was the co-founder and a CTO of Bravolol Ltd., Hong Kong, a leading language learning mobile application company with over 100 million users, and listed as the top two language education platforms globally. He has authored or co-authored around

50 papers published and presented in prestigious conferences and journals, such as the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE INTERNET OF THINGS JOURNAL, *ACM Transactions on Multimedia Computing Communication and Application*, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, *IEEE Communications Magazine*, IEEE NETWORK, HICSS, ACM MobiCom, and WWW. His current research interests include mobile cyber-physical systems, crowdsensing, social networks, and cloud computing.

Dr. Hu has been serving as the Lead Guest Editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and *Wireless Communications and Mobile Computing*.



Zhaolong Ning (M'14) received the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China.

He was a Research Fellow with Kyushu University, Fukuoka, Japan. He is an Associate Professor with the School of Software, Dalian University of Technology, Dalian, China. His current research interests include Internet of Things, edge computing, and vehicular networks.



Edith C.-H. Ngai (S'02–M'07–SM'12) received the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2007.

She is currently an Associate Professor with the Department of Information Technology, Uppsala University, Uppsala, Sweden. She has been a Visiting Researcher with Ericsson Research, Stockholm, Sweden, from 2015 to 2017. She was a Post-Doctoral Fellow with Imperial College London, London, U.K. She has also been a Visiting Researcher with Simon Fraser University, Burnaby, BC, Canada, Tsinghua University, Beijing, China, and the University of California at Los Angeles, Los Angeles, CA, USA. She is a Project Leader of the National Project, GreenIoT from 2014 to 2017, for open data and sustainable development in Sweden. Her current research interests include Internet-of-Things, mobile cloud computing, network security and privacy, and smart cities and urban computing.

Dr. Ngai was a recipient of the VINNMER Fellow awarded by the Swedish governmental agency VINNOVA in 2009, the Best Paper Runner-Up Award of IEEE IWQoS 2010 and ACM/IEEE IPSN 2013 for her co-authored papers. She served as the TPC Co-Chair of IEEE SmartCity 2015, IEEE ISSNIP 2015, and the ICNC 2018 Network Algorithm and Performance Evaluation Symposium. She is an Associate Editor of IEEE ACCESS and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. She is a Senior Member of the ACM.



Li Zhou received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China in 2009, 2011, and 2015, respectively.

From 2013 to 2014, he was a Visiting Scholar with the University of British Columbia, Vancouver, BC, Canada. He is currently an Assistant Professor with the College of Electronic Science, NUDT. His research contributions have been published and presented in over 20 prestigious journals and conferences, such as the IEEE TRANSACTIONS ON

VEHICULAR TECHNOLOGY, *Ad Hoc Networks*, WInnComm 2017, IEEE INFOCOM 2015, and IEEE GLOBECOM 2014. His current research interests include software defined radios, software defined networks, and heterogeneous networks.

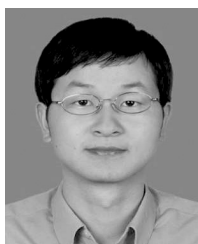
Dr. Zhou served as a TPC member of IEEE CIT 2017, a Keynote Speaker of ICWCNT 2016, and the Co-Chair of ITA 2016.



Jibo Wei received the B.S. and M.S. degrees in electronic engineering from the National University of Defense Technology (NUDT), Changsha, China, in 1989 and 1992, respectively, and the Ph.D. degree in electronic engineering from Southeast University, Nanjing, China, in 1998.

He is currently the Director and a Professor with the Department of Communication Engineering, NUDT. His current research interests include wireless network protocol and signal processing in communications, more specially, the areas of MIMO, multicarrier transmission, cooperative communication, and cognitive network.

Dr. Wei is a member of the IEEE Communication Society and IEEE VTS. He is also an Editor of *Journal on Communications* and a Senior Member of the China Institute of Communications and Electronics.



Jun Cheng received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2006.

He is currently a Professor and the Founding Director of the Laboratory for Human Machine Control, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He has authored or co-authored about 110 articles. His current research interests

include computer visions, robotics, and machine intelligence and control.



Bin Hu (M'10–SM'15) is currently a Professor and the Dean of the School of Information Science and Engineering, Lanzhou University, Lanzhou, China, an Adjunct Professor with Tsinghua University, Beijing, China, and a Guest Professor with ETH Zurich, Zürich, Switzerland. He has been funded as a PI by the Ministry of Science and Technology, National Science Foundation China, European Framework Programme 7, EPSRC, and HEFCE U.K. He has authored or co-authored over 200 papers in peer-reviewed journals, conferences, and book chapters, including *Science* (Suppl.), the *Journal of Alzheimer's Disease*, IEEE TRANSACTIONS, IEEE INTELLIGENT SYSTEMS, AAAI, BIBM, EMBS, CIKM, and ACM SIGIR.

Prof. Hu has served as the Guest Editor of *Science* in the Special Issue on "Advances in Computational Psychophysiology," an Associate Editor of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, *Brain Informatics*, *IET Communications*, *Cluster Computing*, *Wireless Communications and Mobile Computing*, and *Security and Communication Networks* (Wiley). He is the Co-Chair of the IEEE SMC TC on Cognitive Computing, a Member-at-Large of ACM China, and the Vice President of the International Society for Social Neuroscience (China Committee). He is an IET Fellow.



Victor C. M. Leung (S'75–M'89–SM'97–F'03) received the B.A.Sc. degree (Hons.) in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977, and the Ph.D. degree in electrical engineering from the Graduate School, UBC, in 1982.

From 1981 to 1987, he was a Technical Staff Senior Member with Microtel Pacific Research Ltd. (currently MPR Teltech Ltd.), Burnaby, BC, Canada, where he specialized in the planning, design, and analysis of satellite communication systems. He also

held a part-time position as a Visiting Assistant Professor with Simon Fraser University, Burnaby, in 1986 and 1987. In 1988, he was a Lecturer with the Department of Electronics, Chinese University of Hong Kong, Hong Kong. He returned to UBC as a faculty member in 1989, where he is currently a Professor and the inaugural holder of the TELUS Mobility Industrial Research Chair with Advanced Telecommunications Engineering, Department of Electrical and Computer Engineering. He is the Director of the Laboratory for Wireless Networks and Mobile Systems and a member of the Institute for Computing, Information, and Cognitive Systems, UBC. He also holds or has held a Guest/Adjunct Professor appointment with Jilin University, Changchun, China, Beijing Jiaotong University, Beijing, China, the South China University of Technology, Guangzhou, China, Hong Kong Polytechnic University, Hong Kong, the Beijing University of Posts and Telecommunications, Beijing, and the Hefei University of Technology, Hefei, China. He has authored or co-authored over 1100 papers in archival journals and international conferences, as well as 40 book chapters. He has also co-authored/co-edited 14 books. His current research interests include wireless networks and mobile systems.

Dr. Leung was a recipient of the APEBC Gold Medal as the Head of the Graduating Class of the Faculty of Applied Science, the 2011 UBC Killam Research Prize, the IEEE Vancouver Section Centennial Award, the 2017 Canadian Award for Telecommunications Research, the 2018 IEEE ComSoc TGCC Distinguished Technical Achievement Recognition Award, the 2017 IEEE ComSoc Fred W. Ellersick Prize for his co-authored paper, the 2017 IEEE Systems Journal Best Paper Award, and the 2018 IEEE ComSoc CSIM Best Journal Paper Award. He is serving on the Editorial Boards of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, and *Computer Communications*. He has previously served on the Editorial Boards of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Wireless Communications Series and Series on Green Communications and Networking), the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE WIRELESS COMMUNICATIONS LETTERS, and the *Journal of Communications and Networks*. He has guest-edited many special journal issues, and served on the Technical Program Committee of numerous international conferences. He has provided leadership to the Technical Program Committees and Organization Committees of many international conferences. He was a Distinguished Lecturer of the IEEE Communications Society from 2009 to 2012. He is a registered member of Engineers and Geoscientists BC, Canada. He is a Fellow of the Academy of Science, Royal Society of Canada, Engineering Institute of Canada, and the Canadian Academy of Engineering, and a Voting Member of the ACM.