A Novel Mobile Edge Network Architecture with Joint Caching-Delivering and Horizontal Cooperation

Yuris Mulya Saputra, Dinh Thai Hoang, Diep N. Nguyen, and Eryk Dutkiewicz

Abstract—Mobile edge caching/computing (MEC) has been emerging as a promising paradigm to provide ultra-high rate, ultra-reliable, and/or low-latency communications in future wireless networks. In this paper, we introduce a novel MEC network architecture that leverages the optimal joint caching-delivering with horizontal cooperation among mobile edge nodes (MENs). To that end, we first formulate the content-access delay minimization problem by jointly optimizing the content caching and delivering decisions under various network constraints (e.g., network topology, storage capacity and users' demands at each MEN). However, such strongly mutual dependency between the decisions creates a nested dual optimization that is proved to be NP-hard. To deal with it, we propose a novel transformation method to transform the nested dual problem to an equivalent mixed-integer nonlinear programming (MINLP) optimization problem. Then, we design a centralized solution using an improved branch-and-bound algorithm with the interior-point method to find the joint caching and delivering policy which is within 1% of the optimal solution for small problem instances. Since the centralized solution requires the full network topology and information from all MENs, to make our solution scalable, we develop a distributed algorithm which allows each MEN to make its own decisions based on its local observations. Extensive simulations demonstrate that the proposed solutions can reduce the total average delay for the whole network up to 40% compared with other current caching policies. Furthermore, the proposed solutions also increase the cache hit ratio for the network up to 4 times, thereby dramatically reducing the traffic load on the backhaul network.

Index Terms—Mobile edge caching, horizontal cooperative caching, joint caching-delivering, branch-and-bound, delay, interior-point.

---

1 INTRODUCTION

THE proliferation of smart devices (e.g., smartphones, tablets, wearable devices) and a massive demand for emerging services (e.g., IoT, mobile videos, augmented and virtual reality applications) will lead to an explosion of mobile data traffic in the near future [1]. Moreover, new services/applications (e.g., critical missions, safety, or real-time entertainment) often require much lower latency (e.g., in order of 10 milliseconds or less) and/or higher reliability (e.g., 99.999%). To cope with this ever-increasing traffic demand and much more stricter quality of service, mobile edge caching/computing (MEC) [2], [3], [4] has been proposed as one of the most effective solutions. The key idea of an MEC network is to distribute well-received contents as well as computing resources closer to the mobile users by deploying servers at the "edge" of the network, referred to as mobile edge nodes/servers (MENs). By doing so, the MEC network allows mobile users, instead of downloading contents from cloud/content servers (CSs), to access contents and resources from nearby MENs. As such, MEC helps to reduce the service latency and mitigate the network congestion on the backhaul link [5]. The deployment of MEC network also brings other significant benefits to the mobile users, e.g., reliable wireless connections, high speed data transfer, and low energy consumption, and reduces

---

- *The authors are with the School of Electrical and Data Engineering, University of Technology Sydney, Australia.*
  E-mail: yurismulya.saputra@student.uts.edu.au,
  {hoang.dinh, diep.nguyen, eryk.dutkiewicz}@uts.edu.au.

expensive operational as well as upgrading costs on the backhaul link for the MEC network providers.

Despite of the advantages, the development of MEC networks faces several inherent challenges, e.g., diverse users' demands, small coverage, and limited storage capacity of each MEN [6]. To overcome these issues, cooperative caching has been introduced recently. Cooperative caching is a method which leverages the distribution of contents [7] through the cooperation among MENs and provides efficient workload distribution in a hierachical architecture [8]. In [9], the authors introduce an optimal cooperative content placement strategy aiming to maximize the total hit rate of contents for a heterogeneous cellular network (referred to as the network hit rate). To achieve the optimal policy, the scheme first caches contents randomly at different layers, e.g., macro, pico, and femto nodes. Then, a stochastic geometry is deployed to evaluate the request hit rate/probability for the network. Based on the request hit rate evaluation together with the caching storage capacities of MENs, the maximum network hit rate then can be derived. Similar to [7], the authors in [10] propose a distributed cooperative caching architecture consisting of several local nodes such as small cells or base stations to reduce the content delivery delay. In this work, the network providers collaborate together through sharing their MENs to avoid long service delay from the content servers. For example, users of network provider X can download contents from MENs of network provider Y. In [11], a collaborative caching design based on a tree architecture is considered. In this way, if a requested content is not cached at the leaf node server, this node can

download content from its directly connected parent node. Based on this architecture, the authors introduce a two-level cooperative caching group model to minimize the total bandwidth cost for the system. The work is then extended in [12] with network topology taken into account.

To minimize the service delay, most of the above cooperative caching solutions focus only on content placement strategies (e.g., [5], [11], and [12]), but not on how content should be delivered/routed. Among few works that investigate both caching and delivering strategies, the authors in [13] introduce dual objective approaches to optimize cooperative caching and delivering policy for heterogeneous cellular networks (HetNets) with femtocells and D2D communications. As such, after obtaining the optimal caching strategy, the authors adopt an optimal content delivering strategy to find the best MEN to serve if there are more than one MENs caching the same requested content. Nonetheless, in this work, caching placement and delivery optimizations are separately optimized, leading to a suboptimal solution. A similar suboptimal solution is also found in [14].

Moreover, for all the aforementioned works, the authors do not consider the cooperation among MENs (referred to as *horizontal cooperation*) in delivering requested contents. MENs are often launched in a close proximity where the horizontal (wireless or even wire) connections among them have much higher speed than that of the backhaul link or those from MENs to the CSs. This fact, if leveraged properly, facilitates the cooperation amongst MENs in jointly caching and delivering contents to not only reduce the service delay for mobile users but also improve the caching effectiveness (by minimizing the backhaul traffic to the CSs).

In this paper, we introduce an optimal **JO**int **C**ooperative c**A**ching and **D**elivering framework (referred to as JOCAD) that enables MENs to cooperate in caching and delivering contents to mobile users. Specifically, we first propose a novel MEC network architecture in which MENs can be connected with each other directly (with high-speed connections). Then, given the content demand distributions (referred to as frequency-of-accesses) at MENs, various data sizes, diverse MENs' storage capacities, and network topology (i.e., connections among the MENs and their bandwidth), we aim to jointly address two essential questions: (1) how to place contents at the MENs efficiently and (2) how to choose the best routes to deliver requested contents.

Due to the strongly mutual dependency between content placement and delivery decisions, the joint content caching and delivering optimization problem turns to be intractable. In particular, where to cache contents will be influenced by the delivering decisions. Likewise, the decision to deliver the contents will be impacted by the content placement strategy. Consequently, this inter-dependency gives rise to a nested dual optimization problem, that we prove to be NP-hard. To tackle it, we propose a novel transformation method to transform the nested dual problem to an equivalent mixed-integer nonlinear programming (MINLP) optimization problem. By exploiting the unique structure of this MINLP problem, we propose a centralized cooperative caching-delivering solution (referred to as centralized solution) using an improved branch-and-bound algorithm with the interior-point method (iBBA-IPM) to find the joint caching and delivering policy which is within 1% of the optimal solution for small problem instances.

Nevertheless, the centralized solution requires a full network topology and the information from all MENs, making it prohibitively costly for large-scale systems, especially when the number of contents and MENs is huge. Moreover, the centralized solution also requires coordination overhead from all MENs. To make our solution scalable and reduce the complexity and communication overheads among MENs, we develop a distributed cooperative caching-delivering algorithm (referred to as distributed solution) which allows each MEN to make its own decisions based on its local observations, e.g., connections to their neighbors. Extensive simulations demonstrate that the proposed solutions can reduce the total average delay for the whole network up to 40% compared with the most FoA policy, and up to 25% compared with locally optimal caching policy (i.e., without collaboration). Furthermore, the proposed solutions also increase the cache hit ratio for the network up to 4 times, thereby dramatically reducing the traffic load on the backhaul network. The major contributions are summarized as follows:

- We design the JOCAD framework with mutual dependency that utilizes the direct horizontal cooperation among MENs to minimize the total delay for the MEC network and traffic load on the backhaul links. We show that the optimization problem is NP-hard.
- We then propose a novel transformation method to transform the intractable original optimization to an equivalent MINLP problem which we can adopt effective mathematical tools to address.
- We develop the centralized solution using iBBA-IPM to find the joint caching and delivering policy for the MINLP problem which can achieve within 1% of the optimal solution.
- We design the distributed solution to reduce the complexity of the centralized solution. We demonstrate that the distributed solution can achieve the performance close to that of the centralized solution.
- We conduct extensive simulations to evaluate the efficiency of the proposed framework and solutions. These results also provide insightful information to help the MEC service providers tradeoff between the quality of service, e.g., delay, and the implementation costs, e.g., the number of MENs and storage capacity.

The rest of this paper is organized as follows. Section 2 describes the proposed network architecture and model. Section 3 discusses the problem formulation and the transformation method. The centralized and distributed solutions are in Section 4 and Section 5, respectively. A case study is given in Section 6, and then Section 7 shows the simulation results. Conclusions are then drawn in Section 8.

2 MOBILE EDGE NETWORK ARCHITECTURE WITH HORIZONTAL COOPERATION AMONGST MENS

The proposed MEC network architecture with direct horizontal collaboration among the MENs is described in Fig. 1. This architecture adopts a practical MEC model in 5G networks proposed by The European Telecommunications Standards Institute (ETSI) [15]. Each MEN in the network
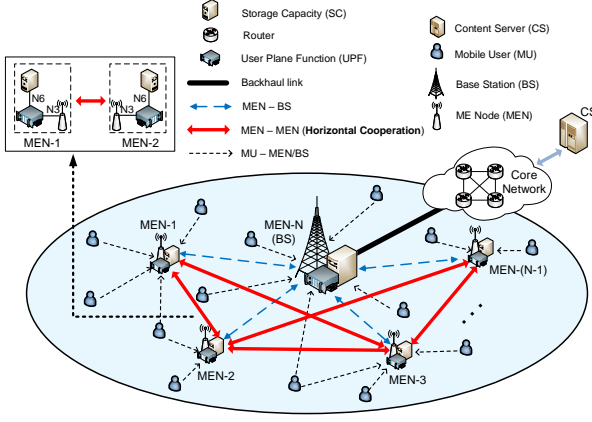
Fig. 1: Proposed MEC network architecture with direct horizontal cooperation.

serves a set of mobile users in its coverage area and it can communicate with its nearby MENs (the BS is also considered as an MEN in the network) using either wireless (e.g., Wi-Fi) or wire connections. Additionally, each MEN is equipped with finite storage capacity to cache popular contents. In this case, each MEN can implement periodic updating strategies of the high-demand content placement within a day at off-peak hours, e.g., at night, to reduce traffic loads during peak hours [3], [16]. To route data traffic, e.g., content requests and content downloads, from/to the mobile users or other MENs, each MEN also utilizes a user plane function (UPF). This UPF is connected to the MEN and the storage capacity via N3 and N6 interfaces, respectively [15]. Moreover, the mobile users can move from one MEN's coverage to another MEN's area to create connections for content requests.

When a content request is sent to an MEN, if the content is cached locally at the MEN, it will send the content to the user immediately through the UPF. If the content is not stored locally at the MEN but at its neighboring MENs (i.e., directly connected to the MEN), the MEN will download the content from the node which has the lowest delivery delay before sending the requested content to the user. Otherwise, the MEN will download the content from the CS via the BS[1]. In this case, the BS requires to download the content from the CS through the 5G core network (CN) [17]. Note that if the MEN is the BS, it will check from MENs in the network first, and if there is no MEN storing this content, it will download the content from the CS via the 5G CN. In this way, the proposed model can leverage the direct horizontal cooperation among MENs to reduce content-access delay for the users as well as traffic load on the backhaul network.

Let $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$ denote the set of the MENs. The BS is denoted by MEN-N. Each MEN-n has storage capacity and the number of mobile users in its coverage area denoted by s_n and U_n, respectively. B represents the bandwidth between the BS and the CS. We also define $l_n^m, \forall n, m \in \mathcal{N}, n \neq m$ as the bandwidth between MEN-

---

1. In practice, the bandwidth between the BS and CS is usually much higher than among MENs [18]. Thus, downloading the content from the CS via the BS will have lower delay than getting it through two MEN-MEN hops.

n (note that $B \gg l_n^m$) and MEN-m, and $b_n^u, \forall n \in \mathcal{N}$ as the allocated bandwidth of a user u at MEN-n. Furthermore, we define $\mathcal{I} = \{1, \ldots, i, \ldots, I\}$ as the set of contents. Contents may have diverse data sizes denoted by $\mathcal{C} = \{c_1, \ldots, c_i, \ldots, c_I\}$. We denote frequency-of-access (FoA) of content i at MEN-n as f_n^i.

3 JOINT COOPERATIVE CACHING AND DELIVERING OPTIMIZATION PROBLEM

In this section, we first formulate the joint cooperative caching and delivering optimization problem to minimize the total average delay for the MEC network under various network constraints as aforementioned. This nested dual optimization problem is shown to be NP-hard. We then propose a method to transform the intractable nested dual optimization problem into the equivalent MINLP for which we can develop effective solutions.

3.1 Decision Variables and Problem Analysis

We define $\mathbf{x} \stackrel{\text{def}}{=} [\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N]^T$, where $\mathbf{x}_n = [x_n^1, \ldots, x_n^i, \ldots, x_n^I]$ and $x_n^i \in \{0, 1\}$, as the binary decision vector of the MENs. The variable x_n^i is defined as follows:

$$x_n^i = \begin{cases} 1, & \text{if content } i \text{ is cached at MEN-}n, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

When a user u at MEN-n requests a content i, the following three cases are considered.

3.1.1 Case 1: MEN-n has the requested content
In this case, $x_n^i = 1$, and thus the delay to download the content will be

$$d_{case1}^* \stackrel{\text{def}}{=} x_n^i d_\alpha, \tag{2}$$

where $d_\alpha = \frac{c_i}{b_n^u}$.

3.1.2 Case 2: MEN-n does not have the requested content but at least one of its directly connected MENs caches this content
In this case, $x_n^i = 0$. We denote $\mathcal{M}_n^i$ as the set of MENs which are directly connected to MEN-n and store content i. Because there is at least one MEN in $\mathcal{M}_n^i$ caching the content i, we can derive:

$$x_{m\dagger}^i \stackrel{\text{def}}{=} \prod_{m \in \mathcal{M}_n^i} (1 - x_m^i) = 0, \tag{3}$$

where

$$x_m^i = \begin{cases} 1, & \text{if content } i \text{ is cached at MEN-}m, \\ & m \neq n, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

As such, the MEN-n will download content i from one node which has the lowest delivery time, and thus the delay to download the content in this case will be:

$$d_{case2}^* \stackrel{\text{def}}{=} (1 - x_n^i)(1 - x_{m\dagger}^i)(d_\alpha + d_\beta), \tag{5}$$

where $d_\beta \stackrel{\text{def}}{=} \min_m \left([x_m^i + (1 - x_m^i)V] \frac{c_i}{l_n^m} \right)$, and V is a very large constant number. Here, d_β represents the optimization problem to find an MEN-m which contains requested content i and has the lowest delivery time. In addition, the V ensures that MENs without containing the requested content i (i.e., $x_m^i = 0$) are practically ignored.

It is worth noting that our model is designed to take the full advantage of direct horizontal cooperation among MENs. In fact, MENs are often deployed in a close proximity area. As such, the wire or wireless links among MENs are often in place [10], [19] and usually much faster than the links from MENs to the BS. In particular, if one of its directly connected MENs stores content i, MEN-n will download this content from one of these nodes instead of trying to download the content from the CS or from another MEN in the network via the BS. This strategy is to minimize the latency, the traffic on the backhaul network as well as inside the MEC network.

Note that in the case MEN-n is the BS, i.e., $n = N$, only *Case 1* and *Case 2* can occur as the BS is directly connected to the CS. We also do not consider direct connections between MENs and the CS. This is due to the fact that the direct connection between an MEN and the CS often has a very low bandwidth capacity [18]. Another reason is that deploying direct connections between MENs and the CS will cause significant infrastructure deployment and operational costs for the MEC service providers [14], [19], [20].

3.1.3 Case 3: MEN-n and all of its directly connected nodes do not have the requested content

In this case, we have $x_n^i = 0$ and $x_{m^\dagger:r}^i = 1$. Then, there are two possibilities. First, there is no MEN in the network storing the content. Then, the content will be downloaded from the CS via the BS, and the delay in this case will be

$$d_{case3}^* \stackrel{\text{def}}{=} (1 - x_n^i) x_{m^\dagger}^i d_\delta, \tag{7}$$

where $d_\delta = \left(d_\alpha + \frac{c_i}{l_n^N} + \frac{c_i}{B} \right)$. Second, if there is at least one MEN in the network that is not directly connected to MEN-n, say MEN-m (with some abusing of notation), storing this content, MEN-n will download the content from either the CS via the BS or from MEN-m via the BS or any intermediate MEN (whichever has lower delivery delay). However, in practice, as aforementioned the bandwidth between the BS and CS is usually much higher than among MENs [18], and thus the content will be downloaded from the CS in the second case. Consequently, the delay to download the content in *Case 3* will be d_δ.

3.2 Problem Formulation

Based on the aforementioned analysis, we then formulate the joint cooperative caching and delivering optimization problem ($\mathbf{P}_1$) to minimize the total average delay of the MEC network as follows:

$$(\mathbf{P}_1) \quad \min_{\mathbf{x}} F(\mathbf{x}), \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{I} x_n^i c_i \leq s_n, \forall n \in \mathcal{N}, \tag{9}$$

$$x_n^i, x_m^i \in \{0, 1\}, \forall n, m \in \mathcal{N}, n \neq m, \forall i \in \mathcal{I}, \tag{10}$$

where the objective function $F(\mathbf{x})$ is defined in Eq. (6) as the total average delay at all MENs for all contents in the MEC network. The constraints (9) guarantee that the total size of all cached contents does not exceed the storage capacity of each MEN. Additionally, the constraints (10) specify that caching decision variables are binary. Based on ($\mathbf{P}_1$), our optimization problem is considered to be a *nested dual binary nonlinear programming*. In particular, binary decision variables are multiplied and used in both inner and outer minimization functions: (1) the outer level (OL) is the main objective function $F(\mathbf{x})$ to minimize the total average delay of the network and (2) the inner level (IL) is the optimal delivering decision problem, i.e.,

$$F_{IL}(\mathbf{x}) = d_\beta = \min_{m} \left([x_m^i + (1 - x_m^i)V] \frac{c_i}{l_n^m} \right). \tag{11}$$

The IL will try to find a directly connected node, i.e., MEN-m, which minimizes the delivery time to a requesting mobile user as described in *Case 2*. Furthermore, the OL and IL functions have a unique feature, i.e., *mutual-dependency*. Specifically, on the one hand, where to cache contents (at MENs) will influence on how to deliver the contents. On the other hand, how to deliver contents (depending on the network topology) will impact on how/where contents should be cached at MENs. In this way, to address the OL problem, we need to solve the IL problem by obtaining the optimal value of m, i.e., which MENs to deliver. Furthermore, it is important to note that we cannot solve the IL problem to find the optimal values of m separately because the optimal values of m strongly depend on the variables $\mathbf{x}$ (i.e., which contents are cached at which MENs). Consequently, to minimize the total average delay for the network, we need to simultaneously address the joint caching and delivering optimization problem. In Lemma 1, we show that the optimization problem in Eq. (8) is an NP-hard problem.

Lemma 1. *The nested dual optimization ($\mathbf{P}_1$) is NP-hard.*

Proof. See Appendix A. ☐

3.3 Problem Transformation

In ($\mathbf{P}_1$), we need to simultaneously optimize the content caching and delivering decisions. However, as discussed above, the decisions to cache and to deliver the contents are mutual dependent. Hence, the optimization problem is unfortunately not a typical bilevel optimization [21] in which the inner and outer decision variables are independent. As such, conventional techniques used in the bilevel optimization are not applicable. To address this problem, we propose a novel method to transform the intractable nested dual optimization problem into an equivalent mixed-integer nonlinear programming (MINLP) optimization problem which can be solved by using effective mathematical tools.

Specifically, we introduce a vector of functions $Q(\mathbf{x})$ as well as vectors of auxiliary variables $\mathbf{z}$ and binary variables $\mathbf{y}$ to replace the IL problem $F_{IL}(\mathbf{x})$, and add appropriate constraints to make the transformed problem to be equivalent. In this case, $Q(\mathbf{x})$ represents the IL problem. Particularly, $Q(\mathbf{x}) \stackrel{\text{def}}{=} [Q_n(\mathbf{x}_1), \ldots, Q_n(\mathbf{x}_m), \ldots, Q_n(\mathbf{x}_{M_n})]^T$, where $Q_n(\mathbf{x}_m) \stackrel{\text{def}}{=} [Q_n^1(x_m^1), \ldots, Q_n^i(x_m^i), \ldots, Q_n^I(x_m^I)]$ and $Q_n^i(x_m^i) \stackrel{\text{def}}{=} [x_m^i + (1 - x_m^i)V] \frac{c_i}{l_n^m}$. Then, we define $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n, \ldots, \mathbf{z}_N]^T$ with $\mathbf{z}_n = [z_n^1, \ldots, z_n^i, \ldots, z_n^I]$, where $z_n^i \in \mathbb{R}_0^+$, as the vector of $F_{IL}(\mathbf{x})$ solutions for all contents and MENs. Furthermore, we denote $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m, \ldots, \mathbf{y}_N]^T$ with $\mathbf{y}_m = [y_m^1, \ldots, y_m^i, \ldots, y_m^I], y_m^i \in \{0, 1\}$, and $m \neq n$, as the binary decision vector to help MEN-n, $\forall n \in \mathcal{N}$ to find another directly connected MEN-m, $m \in \mathcal{M}_n^i$, which has the shortest time to deliver requested content i.

$$F(\mathbf{x}) = \sum_{i=1}^{I}\sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(d_{case1}^* + d_{case2}^* + d_{case3}^* \right) \right]$$

$$= \sum_{i=1}^{I}\sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\underbrace{x_n^i d_\alpha}_{Case\ 1} + \underbrace{(1-x_n^i)(1-x_{m\dagger}^i)\left(d_\alpha + \min_m \left([x_m^i + (1-x_m^i)V]\frac{c_i}{l_n^m} \right) \right)}_{Case\ 2} + \underbrace{(1-x_n^i)x_{m\dagger}^i d_\delta}_{Case\ 3} \right) \right] \quad (6)$$

$$= \sum_{i=1}^{I}\sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(x_n^i d_\alpha + (1-x_n^i)\left[(1-x_{m\dagger}^i)(d_\alpha + d_\beta) + x_{m\dagger}^i d_\delta \right] \right) \right]$$

Given the vectors $Q(\mathbf{x})$, $\mathbf{z}$, and $\mathbf{y}$, the new equivalent optimization problem ($\mathbf{P}_2$), known as MINLP problem, can be expressed as follows:

$$(\mathbf{P}_2) \quad \min_{\{\mathbf{x},\mathbf{y},\mathbf{z}\}} F(\mathbf{x},\mathbf{y},\mathbf{z}), \quad (12)$$

s.t. (9)-(10) and

$$z_n^i \geq Q_n^i(x_m^i) - V y_m^i, \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \quad (13)$$

$$\sum_{m \in \mathcal{M}_n^i} y_m^i = M_n^i - 1, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \quad (14)$$

$$y_m^i \in \{0,1\}, \forall m \in \mathcal{N}, \forall i \in \mathcal{I}, \quad (15)$$

$$z_n^i \in \mathbb{R}_0^+, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \quad (16)$$

where

$$F(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=1}^{I}\sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(x_n^i d_\alpha + \right. \right.$$
$$\left. \left. (1-x_n^i)\left[(1-x_{m\dagger}^i)(d_\alpha + z_n^i) + x_{m\dagger}^i d_\delta \right] \right) \right], \quad (17)$$

and M_n^i is the cardinality of the set $\mathcal{M}_n^i$. The constraints (13) represent the condition to select one MEN-m which has the lowest delivery time. In addition, the aim of constraints (14) and (15) are to guarantee that only one variable y_m^i is set to be "0", while the rest of variables are set to be "1". Specifically, $M_n^i - 1$ indicates that when M_n^i number of directly connected MENs containing content i for MEN-n are considered, we exclude one node which has $y_m^i = 0$ (i.e., the selected MEN-m to deliver the content i). The equivalent transformation is formally stated in Theorem 1.

THEOREM 1. *The nested dual optimization problem ($\mathbf{P}_1$) is equivalent to MINLP optimization problem ($\mathbf{P}_2$).*

Proof. See Appendix B. □

Based on ($\mathbf{P}_2$), we can simplify some parts of the objective function in Eq. (12) as

$$\Omega(x_n^i, x_m^i) \stackrel{\text{def}}{=} x_n^i d_\alpha + (1-x_n^i)(1-x_{m\dagger}^i)d_\alpha \quad (18)$$
$$+ (1-x_n^i)x_{m\dagger}^i d_\delta,$$

$$\Phi(x_n^i, x_m^i) \stackrel{\text{def}}{=} (1-x_n^i)(1-x_{m\dagger}^i), \quad (19)$$

the constraints (9) as

$$\Theta(x_n^i) \stackrel{\text{def}}{=} \sum_{i=1}^{I} x_n^i c_i - s_n, \quad (20)$$

and the constraints (14) as

$$\Gamma(y_m^i) \stackrel{\text{def}}{=} \sum_{m \in \mathcal{M}_n^i} y_m^i - (M_n^i - 1). \quad (21)$$

Then, we have

$$F(\mathbf{x},\mathbf{y},\mathbf{z}) =$$
$$\sum_{i=1}^{I}\sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(x_n^i, x_m^i) + \Phi(x_n^i, x_m^i) z_n^i \right) \right], \quad (22)$$

and the MINLP problem ($\mathbf{P}_2$) becomes

$$(\mathbf{P}_3) \quad \min_{\{\mathbf{x},\mathbf{y},\mathbf{z}\}} F(\mathbf{x},\mathbf{y},\mathbf{z}), \quad (23)$$

s.t. (10), (15)-(16), and

$$\Theta(x_n^i) \leq 0, \forall n \in \mathcal{N}, \quad (24)$$

$$Q_n^i(x_m^i) - V y_m^i - z_n^i \leq 0, \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \quad (25)$$

$$\Gamma(y_m^i) = 0, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}. \quad (26)$$

4 CENTRALIZED COOPERATIVE CACHING-DELIVERING SOLUTION

To address ($\mathbf{P}_3$), in this section, we introduce an improved branch-and-bound algorithm (iBBA) which can effectively achieve the near-optimal joint caching and delivering policy for the whole network.

4.1 iBBA with Interior-Point Method

We adopt branch-and-bound algorithm (BBA) [22] which reduces the time complexity and leverages the characteristics of binary variables to find the final solution of ($\mathbf{P}_3$). The BBA has been commercially popular (in CPLEX by IBM [23] or in SBB by GAMS [24]) and it can work very effectively to solve integer programming problems. Nonetheless, to solve the non-linearity and continuous relaxation of ($\mathbf{P}_3$), we employ the interior-point method (IPM) which has polynomial time complexity [25]. The IPM is considered to be robust and efficient method to address function evaluations and second derivative information for large-scale and sparse nonlinear problem [26] (as implemented in IPOPT [27] and in KNITRO [28]. Overall, both BBA and IPM can be integrated to handle binary and continuous variables of ($\mathbf{P}_3$) efficiently.

Given the aforementioned BBA and IPM, we develop a two-level approach to effectively address ($\mathbf{P}_3$). In particular, the BBA is used as outer-level approach which creates continuous nonlinear subproblems by relaxing binary constraints of ($\mathbf{P}_3$), and then the IPM is applied as inner-level approach to solve the subproblems. In this way, the optimal solutions of subproblems attained from the IPM may not be integral (some variables do not have binary values). Thus, the BBA is then used again to find the feasible (integral) solutions until the final solution of ($\mathbf{P}_3$) is found.

To find the final solution of ($\mathbf{P}_3$) using the BBA, we recall binary variable vectors $\mathbf{x}$ and $\mathbf{y}$, and non-integer variable

vector $\mathbf{z}$. Given I number of contents and N number of MENs, the number of variables of vector $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are $J_{\mathbf{x}} = J_{\mathbf{y}} = J_{\mathbf{z}} = I \times N$. The BBA first relaxes all binary variables x_n^i, x_m^i, and y_m^i of ($\mathbf{P}_3$) into continuous variables at the root problem ($\mathbf{RP}$). In the ($\mathbf{RP}$), the relaxed binary variables are bounded by $0 \leq x_n^i, x_m^i, y_m^i \leq 1$. Then, the ($\mathbf{RP}$) can be expressed as follows:

$$(\mathbf{RP}) \quad \min_{\{\mathbf{x},\mathbf{y},\mathbf{z}\}} F_{\mathbf{RP}}(\mathbf{x},\mathbf{y},\mathbf{z}), \tag{27}$$

$$\text{s.t. (24)-(26) and}$$
$$0 \leq x_n^i, x_m^i, y_m^i \leq 1, z_n^i \in \mathbb{R}_0^+,$$
$$\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \tag{28}$$

where $F_{\mathbf{RP}}(\mathbf{x},\mathbf{y},\mathbf{z}) = F(\mathbf{x},\mathbf{y},\mathbf{z})$ in Eq. (22).

If all variables x_n^i, x_m^i, and y_m^i are binary values (i.e., feasible solution is obtained), the algorithm will stop immediately. Otherwise, it will break the ($\mathbf{RP}$) into subproblems ($\mathbf{SP}$s), i.e., branch problems. In this case, the ($\mathbf{SP}$)s fix one of relaxed decision variables (i.e., x_n^i, x_m^i, or y_m^i) to be "0" at the left branch and "1" at the right branch. We denote the fixed decision variable to be $\zeta \in \{x_n^i, x_m^i, y_m^i\}$ with $F_{\mathbf{SP}}(\mathbf{x},\mathbf{y},\mathbf{z}) = F_{\mathbf{RP}}(\mathbf{x},\mathbf{y},\mathbf{z})$.

In the BBA, each iteration does not need to search all branch problems (or leaves). Instead, ($\mathbf{SP}$) is pruned if one of these two following conditions is met: (1) $\Psi^c > \beta_U$ or (2) $\xi < \tau$. In this case, Ψ^c and β_U refer to the current total average delay and the upper bound of the total average delay. Meanwhile, ξ and τ represent the integrality between relaxed and rounded decision variables G, where $G = \{x_n^i, x_m^i, y_m^i\}$, and integrality gap threshold, respectively. Then, the final solutions $\hat{x}_n^i$, $\hat{x}_m^i$, $\hat{y}_m^i$, and $\hat{z}_n^i$ with $\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$ are obtained if the total average delay $\hat{\Psi} = \Psi \leq \Psi^c$ for all x_n^i, x_m^i, y_m^i, and z_n^i in the problem, where Ψ refers to the incumbent total average delay. To guarantee that the near-optimal solution exists, we set an optimality tolerance η as a non-negative value. Specifically, $\hat{x}_n^i$, $\hat{x}_m^i$, $\hat{y}_m^i$, and $\hat{z}_n^i$ are $\eta-$optimal when the total average delay $\hat{\Psi}$ is tightened within the bounds and the difference between upper bound β_U and lower bound β_L is less than the optimality tolerance as expressed below:

$$\beta_L \leq \hat{\Psi} \leq \beta_U \text{ and } \beta_U - \beta_L \leq \eta. \tag{29}$$

While the BBA handles the feasibility of the ($\mathbf{SP}$)s and the optimality of ($\mathbf{P}_3$), the interior-point method (IPM) is used to solve the nonlinear continuous relaxation of the ($\mathbf{SP}$)s. In particular, we derive an interior-point subproblem ($\mathbf{IP}$) from the ($\mathbf{SP}$) as the approximated problem with additional logarithmic barrier functions and non-negative slack variables to eliminate the inequality operators such that:

$$F_{\mathbf{IP}_\gamma}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4) =$$
$$F_{\mathbf{SP}}(\mathbf{x},\mathbf{y},\mathbf{z}) - \gamma \bigg[\sum_{j_1=1}^{N} \log \sigma_1^{j_1} + \sum_{j_2=1}^{M} \log \sigma_2^{j_2}$$
$$+ \sum_{j_3=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \log \sigma_3^{j_3} + \sum_{j_4=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \log \sigma_4^{j_4} \bigg], \tag{30}$$

and

$$(\mathbf{IP}) \quad \min_{\substack{\mathbf{x},\mathbf{y},\mathbf{z}, \\ \boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4}} F_{\mathbf{IP}_\gamma}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4), \tag{31}$$

$$\text{s.t.} \quad \Theta(x_n^i) + \sigma_1^{j_1} = 0, \forall n, j_1 \in \mathcal{N}, \tag{32}$$
$$Q_n^i(x_m^i) - V y_m^i - z_n^i + \sigma_2^{j_2} = 0, \forall j_2 \in [1, M],$$
$$\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \tag{33}$$
$$\Gamma(y_m^i) = 0, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{34}$$
$$G - \sigma_3^{j_3} = 0, G + \sigma_4^{j_4} = 1,$$
$$\forall j_3, j_4 \in [1, J_{\mathbf{x}} + J_{\mathbf{y}}], \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \tag{35}$$
$$z_n^i \in \mathbb{R}_0^+, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{36}$$

where $M \stackrel{\text{def}}{=} \sum_{i=1}^{I} \sum_{n=1}^{N} M_n^i$, $\gamma > 0$ is the barrier parameter [29], $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2 \in \mathbb{R}_0^+$ are slack variables in the inequality constraints (24) and (25), respectively, and $\boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4 \in \mathbb{R}_0^+$ are slack variables for lower and upper bounds of G in the inequality constraints (28), respectively. In this case, we define $\boldsymbol{\sigma}_1 \stackrel{\text{def}}{=} [\sigma_1^1, \ldots, \sigma_1^{j_1}, \ldots, \sigma_1^N]$, $\boldsymbol{\sigma}_2 \stackrel{\text{def}}{=} [\sigma_2^1, \ldots, \sigma_2^{j_2}, \ldots, \sigma_2^M]$, $\boldsymbol{\sigma}_3 \stackrel{\text{def}}{=} [\sigma_3^1, \ldots, \sigma_3^{j_3}, \ldots, \sigma_3^{J_{\mathbf{x}}+J_{\mathbf{y}}}]$, and $\boldsymbol{\sigma}_4 \stackrel{\text{def}}{=} [\sigma_4^1, \ldots, \sigma_4^{j_4}, \ldots, \sigma_4^{J_{\mathbf{x}}+J_{\mathbf{y}}}]$. To make $F_{\mathbf{IP}_\gamma}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4)$ equal to the minimum $F_{\mathbf{SP}}(\mathbf{x},\mathbf{y},\mathbf{z})$, we need to update γ in decreasing order for each iteration such that it converges to zero. To solve the approximated problem $F_{\mathbf{IP}_\gamma}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4)$, a conjugate gradient technique [30] is adopted to minimize a quadratic approximation of the approximated problem at each step. Let $\mathcal{L}(\mathbf{p},\boldsymbol{\sigma},\boldsymbol{\lambda})$ denote the Lagrangian expression containing the objective function in Eq. (31) along with the corresponding constraints (32)-(35) and a vector of non-negative Lagrangian multipliers $\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ contains $\lambda_1^{j_1}, \forall j_1 \in \mathcal{N}, \lambda_2^{j_2}, \forall j_2 \in [1, M], \lambda_3^{j_3}, \forall j_3 \in [1, J_{\mathbf{x}} + J_{\mathbf{y}}], \lambda_4^{j_4}, \forall j_4 \in [1, J_{\mathbf{x}} + J_{\mathbf{y}}]$, and $\lambda_5^{j_5}, \forall j_5 \in [1, J_{\mathbf{z}}]$. Then, $\mathcal{L}(\mathbf{p},\boldsymbol{\sigma},\boldsymbol{\lambda})$ can be expressed by

$$\mathcal{L}(\mathbf{p},\boldsymbol{\sigma},\boldsymbol{\lambda}) = F_{\mathbf{SP}}(\mathbf{p}) - \gamma \bigg[\sum_{j_1=1}^{N} \log \sigma_1^{j_1} + \sum_{j_2=1}^{M} \log \sigma_2^{j_2}$$
$$+ \sum_{j_3=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \log \sigma_3^{j_3} + \sum_{j_4=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \log \sigma_4^{j_4} \bigg] + \sum_{j_1=1}^{N} \lambda_1^{j_1} \bigg(\Theta(x_n^i) + \sigma_1^{j_1} \bigg)$$
$$+ \sum_{j_2=1}^{M} \lambda_2^{j_2} \bigg(Q_n^i(x_m^i) - V y_m^i - z_n^i + \sigma_2^{j_2} \bigg)$$
$$+ \sum_{j_3=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \lambda_3^{j_3} \bigg(G - \sigma_3^{j_3} \bigg) + \sum_{j_4=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \lambda_4^{j_4} \bigg(G + \sigma_4^{j_4} - 1 \bigg)$$
$$+ \sum_{j_5=1}^{J_{\mathbf{z}}} \lambda_5^{j_5} \Gamma(y_m^i), \tag{37}$$

where $\mathbf{p} = (\mathbf{x},\mathbf{y},\mathbf{z})$ and $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2,\boldsymbol{\sigma}_3,\boldsymbol{\sigma}_4)$. We first need to obtain the Lagrangian multipliers by solving the following equation before using the conjugate gradient:

$$\nabla_{\mathbf{p}}\mathcal{L}(\mathbf{p},\boldsymbol{\sigma},\boldsymbol{\lambda}) = \nabla_{\mathbf{p}}F_{\mathbf{SP}}(\mathbf{p}) + \sum_{j_1=1}^{N} \lambda_1^{j_1} \nabla_{\mathbf{p}}\Theta(x_n^i)$$
$$+ \sum_{j_2=1}^{M} \lambda_2^{j_2} \nabla_{\mathbf{p}} \bigg(Q_n^i(x_m^i) - V y_m^i - z_n^i \bigg) + \sum_{j_3=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \lambda_3^{j_3} \nabla_{\mathbf{p}}G$$
$$+ \sum_{j_4=1}^{J_{\mathbf{x}}+J_{\mathbf{y}}} \lambda_4^{j_4} \nabla_{\mathbf{p}}G + \sum_{j_5=1}^{J_{\mathbf{z}}} \lambda_5^{j_5} \nabla_{\mathbf{p}}\Gamma(y_m^i) = 0. \tag{38}$$

Then, we can use a step size $(\Delta\mathbf{p}, \Delta\boldsymbol{\sigma})$ iteratively to minimize the quadratic approximation such that:

$$\min_{\{\Delta\mathbf{p}, \Delta\boldsymbol{\sigma}\}} F(\Delta\mathbf{p}, \Delta\boldsymbol{\sigma}), \tag{39}$$

$$\text{s.t.} \quad \Theta(x_n^i) + \mathbf{W}_1\Delta\mathbf{p} + \Delta\boldsymbol{\sigma} = 0, \forall n \in \mathcal{N}, \tag{40}$$

$$Q_n^i(x_m^i) - Vy_m^i - z_n^i + \mathbf{W}_2\Delta\mathbf{p} + \Delta\boldsymbol{\sigma} = 0,$$
$$\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \tag{41}$$

$$G + \mathbf{W}_3\Delta\mathbf{p} + \Delta\boldsymbol{\sigma} = 0, G - 1 + \mathbf{W}_4\Delta\mathbf{p} + \Delta\boldsymbol{\sigma} = 0,$$
$$\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \tag{42}$$

$$\Gamma(y_m^i) + \mathbf{W}_5\Delta\mathbf{p} = 0, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{43}$$

where

$$F(\Delta\mathbf{p}, \Delta\boldsymbol{\sigma}) = \nabla_{\mathbf{p}}F_{\mathbf{SP}}(\mathbf{p})^{\mathrm{T}}\Delta\mathbf{p} + \frac{1}{2}\Delta\mathbf{p}^{\mathrm{T}}\nabla_{\mathbf{p}}^2\mathcal{L}(\mathbf{p}, \boldsymbol{\sigma}, \boldsymbol{\lambda})\Delta\mathbf{p}$$
$$+ \gamma\mathbf{e}^{\mathrm{T}}\mathbf{S}^{-1}\Delta\boldsymbol{\sigma} + \frac{1}{2}\Delta\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{A}\Delta\boldsymbol{\sigma}, \tag{44}$$

$\mathbf{e}$ is a vector of ones corresponding to sizes of $\Theta(x_n^i)$, $(Q_n^i(x_m^i) - Vy_m^i - z_n^i)$, and G, $\mathbf{S}$ and $\mathbf{A}$ are the diagonal matrices with the elements of $\boldsymbol{\sigma}$ and $\boldsymbol{\lambda}$ on the diagonal, respectively. Furthermore, $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5)$ represents the Jacobian of the constraint functions $\Theta(x_n^i)$, $(Q_n^i(x_m^i) - Vy_m^i - z_n^i)$, G, and $\Gamma(y_m^i)$.

4.2 Algorithm Complexity and Convergence Analysis of iBBA-IPM

Algorithm 1 describes the pseudocode of the iBBA-IPM solution. First, we evaluate the complexity of the algorithm. The complexity of Algorithm 1 is the lowest when the final solutions $\hat{x}_n^i, \hat{x}_m^i, \hat{y}_m^i, \hat{z}_n^i$ with $\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$ and the final total average delay $\hat{\Psi}$ are found at $(\mathbf{RP})$. However, the complexity becomes higher when $(\mathbf{SP})$s are active at left and right branches of the iBBA-IPM's tree. In particular, we apply a *depth-first search* method by selecting the latest created $(\mathbf{SP})$ as the next $(\mathbf{SP})$ [31] to find the t-th total average delay, i.e., Ψ_t. Suppose that T is the expected number of total average delays $\Psi_t, \forall t \in [1, T]$, searched in the iBBA-IPM's tree, given that the first solution is found at depth ϱ. This T is bounded above and below by linear functions of ϱ to achieve polynomial complexity. This is formally described in Theorem 2.

THEOREM 2. *Given that the first solution is found at depth ϱ in the iBBA-IPM's tree, the expected number of total average delays searched in the iBBA-IPM's tree, i.e., T, follows polynomial complexity at the depth ϱ when T is bounded above and below by linear functions of ϱ.*

Proof. See Appendix C. $\qquad\square$

Next, we show that the iBBA-IPM algorithm converges after a finite number of steps under the optimality tolerance $\eta > 0$ of the optimization problem in $(\mathbf{P}_2)$. This is formally stated in Theorem 3.

THEOREM 3. *The iBBA-IPM algorithm converges after a finite number of steps under the optimality tolerance $\eta > 0$. Specifically, there exists step $\theta_\eta \in \mathbb{N}$ for any $\eta > 0$ such that:*

$$\beta_U^{(\theta_\eta)} - \beta_L^{(\theta_\eta)} \leq \eta, \tag{45}$$

and $\beta_U^{(\theta_\eta)}$ is within the optimality tolerance of the $F(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

Algorithm 1 Centralized Solution with iBBA-IPM

1: $\mathcal{R}_a$: the set of r_a active problems, r_c: the current problem
2: Set $G = \{x_n^i, x_m^i, y_m^i\} \in [0, 1], z_n^i \in \mathbb{R}_0^+, \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$
3: $r_c \leftarrow r_a \leftarrow 1$ /* Set root problem */
4: $\beta_L \leftarrow -\infty, \beta_U \leftarrow +\infty, \Psi \leftarrow \beta_U$
5: Set τ and η /* Integrality gap & optimality tolerance */
6: Solve $(\mathbf{RP})$ for r_c
7: **if** $\forall x_n^i, x_m^i, y_m^i \in \{0, 1\}$ **then**
8:     Store $\hat{x}_n^i, \hat{x}_m^i, \hat{y}_m^i, \hat{z}_n^i$, and $\hat{\Psi}, \forall n, m \in \mathcal{N}, \forall i \in \mathcal{I}$
9:     **return** /* Prune all active problems */
10: **end if**
11: **while** $\mathcal{R}_a \neq \varnothing$ and $\beta_U - \beta_L > \eta$ **do**
12:     $r_c \leftarrow r_a$ /* an $(\mathbf{SP})$ based on the depth-first search */
13:     $r_a \leftarrow r_a - 1$ /* Remove the $(\mathbf{SP})$ from the set */
14:     Solve subproblem $(\mathbf{IP})$ for r_c
15:     **if** $(\mathbf{IP})$ for r_c is infeasible **then**
16:         Prune r_c, **exit**
17:     **else**
18:         Set current G and $z_n^i, \forall n, m \in \mathcal{N}, \forall i \in \mathcal{I}$
19:         Calculate $\xi \leftarrow |G - \text{round}(G)|$
20:         **if** $\xi < \tau$ **then**
21:             **if** $\Psi^c < \Psi$ **then**
22:                 Store current G and $z_n^i, \forall n, m \in \mathcal{N}, \forall i \in \mathcal{I}$
23:                 Set $\Psi \leftarrow \Psi^c$ and $\beta_U \leftarrow \Psi^c$
24:             **end if**
25:             Prune r_c, **exit**
26:         **else**
27:             Choose $\zeta \in G$
28:             $r_a \leftarrow r_a + 2$ /* Add 2 $(\mathbf{SP})$s to the set */
29:             Update $\beta_L \leftarrow \Psi^c$
30:             Update the constraints with $\zeta \leftarrow 0, \zeta \leftarrow 1$, **exit**
31:         **end if**
32:     **end if**
33:     Store $\hat{x}_n^i, \hat{x}_m^i, \hat{y}_m^i, \hat{z}_n^i$, and $\hat{\Psi}, \forall n, m \in \mathcal{N}, \forall i \in \mathcal{I}$
34: **end while**

Proof. See Appendix D. $\qquad\square$

5 DISTRIBUTED COOPERATIVE CACHING-DELIVERING SOLUTION

Despite the fact that the aforementioned centralized solution can find the final solution within 1% to minimize the total average delay for the whole network, it faces two issues. First, the centralized solution requires a centralized computer with the whole network topology and information to be able to solve the nested dual optimization problem. Second, it incurs significant communication overheads among MENs, especially for a large number of MENs. In this section, we introduce a distributed suboptimal solution, which can address these issues. Specifically, under the distributed solution, each MEN first finds the locally optimal caching policy based on the local users' demands and its storage capability. Then, each MEN will communicate with its directly connected nodes to find cooperation through discovering the duplicate contents.

To obtain locally optimal caching policy under the distributed solution, each MEN-n needs to consider three cases when its local user u requests content i:

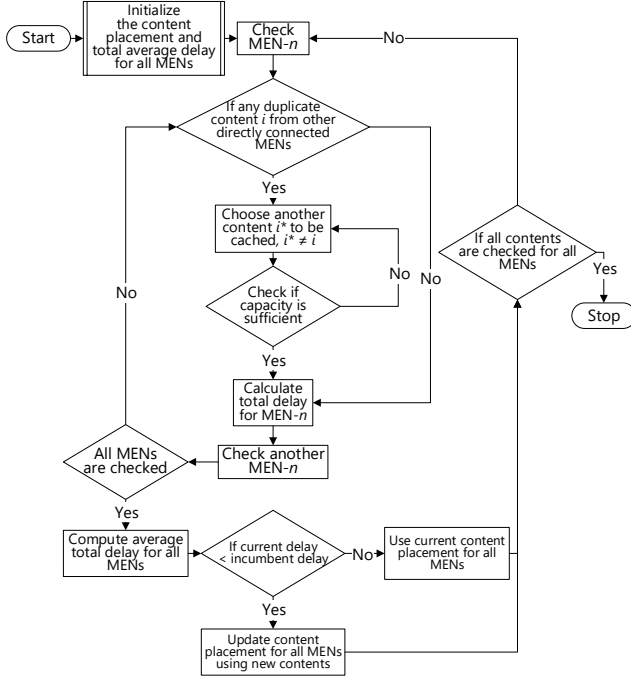Fig. 2: Flowchart of cooperation scheme among MENs for the distributed solution.

- *Case 1*: If MEN-n has the requested content, the delay to deliver the content will be $x_n^i d_\alpha$, where $d_\alpha = \frac{c_i}{b_n^u}$.
- *Case 2*: If MEN-n does not have the requested content but the BS (MEN-N) has the content, the MEN-n will download the content from the BS, and thus the delay will be $(1 - x_n^i)x_N^i d_\beta^\ddagger$, where $d_\beta^\ddagger \overset{\text{def}}{=} \left(d_\alpha + \frac{c_i}{l_n^N}\right)$.
- *Case 3*: If MEN-n and the BS do not have the requested content, the content will downloaded from the CS via the BS, and thus the delay will be $(1 - x_n^i)(1 - x_N^i)d_\delta$, where $d_\delta = \left(d_\alpha + \frac{c_i}{l_n^N} + \frac{c_i}{B}\right)$.

Note that *Case 2* is ignored when users connected directly to the BS. Moreover, since the bandwidth between the BS and CS is usually much higher than that among MENs [18], for *Case 3*, the delay to download the content when users are connected to the BS directly becomes $(1 - x_N^i)d_\delta^\ddagger$, where $d_\delta^\ddagger \overset{\text{def}}{=} \left(d_\alpha + \frac{c_i}{B}\right)$. Based on the aforementioned analysis, we then formulate the locally optimal caching optimization problem, i.e., optimal caching at each node without horizontal cooperation, for MEN-n (where $n \in [1, N-1]$) as $(\mathbf{P}_4)$ with the following equation:

$$(\mathbf{P}_4) \quad \min_{\mathbf{x}} F_n(\mathbf{x}), \tag{46}$$

$$\text{s.t} \sum_{i=1}^{I} x_n^i c_i \le s_n, n \in [1, N-1], \tag{47}$$

$$x_n^i, x_N^i \in \{0, 1\}, n \in [1, N-1], \forall i \in \mathcal{I}, \tag{48}$$

and for the BS as $(\mathbf{P}_5)$ with the following expression:

$$(\mathbf{P}_5) \quad \min_{\mathbf{x}} F_N(\mathbf{x}), \tag{49}$$

$$\text{s.t} \sum_{i=1}^{I} x_N^i c_i \le s_N, \tag{50}$$

$$x_N^i \in \{0, 1\}, \forall i \in \mathcal{I}, \tag{51}$$

where

$$\begin{aligned} F_n(\mathbf{x}) &= \sum_{i=1}^{I} f_n^i \left[\sum_{u=1}^{U_n} \left(x_n^i d_\alpha + \right.\right. \\ &\quad \left.\left. (1 - x_n^i)x_N^i d_\beta^\ddagger + (1 - x_n^i)(1 - x_N^i)d_\delta \right) \right] \\ &= \sum_{i=1}^{I} f_n^i \left[\sum_{u=1}^{U_n} \left(x_n^i d_\alpha + \right.\right. \\ &\quad \left.\left. (1 - x_n^i)\left(x_N^i d_\beta^\ddagger + (1 - x_N^i)d_\delta \right) \right) \right], \end{aligned} \tag{52}$$

and

$$F_N(\mathbf{x}) = \sum_{i=1}^{I} f_N^i \left[\sum_{u=1}^{U_N} \left(x_N^i d_\alpha + (1 - x_N^i)d_\delta^\ddagger \right) \right]. \tag{53}$$

It is shown that the problems $(\mathbf{P}_4)$ and $(\mathbf{P}_5)$ are standard binary linear programming which are generally NP-complete [32]. Nonetheless, the optimization occurs at each MEN only with a much lower number of variables (than the centralized problem). In this way, those problems then can be solved effectively using popular solvers.

After obtaining the locally optimal caching decisions at each MEN, the cooperation among MENs to find the duplicate contents is then carried out. In particular, each MEN-n will consider the duplication of its current cached contents with current ones of its directly connected MEN-m, where $m \in \mathcal{M}_n^i$, at a particular time. The flowchart in Fig. 2 shows how the cooperation among MENs work in this solution. The key idea of this cooperation process is that we need to find the minimum total average delay Ψ in the network by minimizing the duplicate contents among MENs. The process terminates when all candidate contents have been checked at each MEN-n. The complete pseudocode of the distributed solution is shown in Algorithm 2.

The distributed solution algorithm has polynomial complexity. This is shown by the double looping when each content i is checked at each MEN-n. In addition, if an MEN-n has duplicate contents with at least one directly connected MEN-m, the MEN-n will check the rest of the contents (i.e., $I^* = I - 1$). Obviously, the algorithm has polynomial time complexity $O(I \times N \times I^*)$.

6 ILLUSTRATIVE CASE STUDY

To show the efficiency of the proposed solutions using direct horizontal cooperation among MENs, we present an illustrative example in Fig. 3. We use 4 nodes (i.e., 3 MENs and the BS) and 30 available contents with various frequency-of-accesses. The content size is uniformly distributed between 50MB and 200MB. We set the storage capacity at 600MB for MEN-1 to MEN-3 and 1GB for the BS. Suppose that the bandwidth between MEN-n and MEN-m ($\forall n, m \in \mathcal{N}$, and $n \ne m$) is equal. Based on the aforementioned parameter settings, each MEN and the BS can cache up to 4-7 contents.

As seen in Fig. 3, we show caching policies obtained by the locally optimal policy and the proposed solutions (i.e., distributed and centralized solutions). Thanks to horizontal cooperation among MENs in the proposed solutions, MENs can share cached contents to minimize the total average delay for the whole network instead of minimizing the delay

Algorithm 2 Distributed Solution

1: Initialize $x_n^i \in 0, 1, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$ from locally optimal policy problems $(\mathbf{P}_4)$ and $(\mathbf{P}_5)$
2: Check $s_n^*, \forall n \in \mathcal{N}$ /* Current capacity of each MEN */
3: Calculate Ψ /* total average delay of $F_n(\mathbf{x})$ in problem $(\mathbf{P}_4)$ and $F_N(\mathbf{x})$ in problem $(\mathbf{P}_5)$ */
4: **for** $\forall i \in \mathcal{I}$ **do**
5:    **for** $\forall n \in \mathcal{N}$ **do**
6:       **if** $x_n^i = x_m^i = 1, \forall m \in \mathcal{M}_n^i \subset \mathcal{N}, m \neq n$ **then**
7:          Set $s_n^* \leftarrow s_n^* - c_i$
8:          Select $i^*, i^* \neq i, \forall i^* \in \mathcal{I}$
9:          **if** $s_n^* + c_{i^*} \leq s_n$ **then**
10:             Set $x_n^{i^*} \leftarrow 1$ /* Store new candidate content */
11:             Obtain information from MEN-$m, \forall m \in \mathcal{M}_n^i$
12:             Calculate Ψ_n^c /* Current total average delay */
13:          **end if**
14:       **end if**
15:    **end for**
16:    Compute the average of all Ψ_n^c's, $\forall n \in \mathcal{N}$ into Ψ^c
17:    **if** $\Psi^c < \Psi$ **then**
18:       Update $x_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$
19:       $\Psi \leftarrow \Psi^c$
20:    **end if**
21: **end for**
22: Store final $\hat{x}_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$
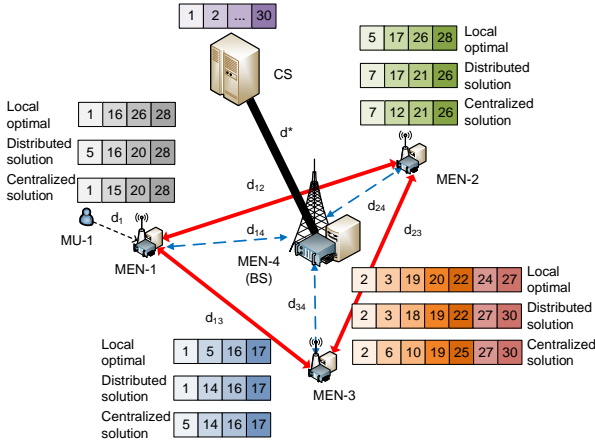23: Calculate final $\hat{\Psi}$ using the solution in Line 22

Fig. 3: An example of caching policies obtained by locally optimal policy, distributed and centralized solutions.

for each individual MEN (as in the locally optimal policy). For example, under the centralized solution, there is no duplicate content cached at all MENs (i.e, each MEN has different set of contents), and thus the number of contents cached at MENs in the network can be maximized (i.e., 19 of 30 available contents are cached in the network). In this way, the centralized solution can find the near-optimal caching and delivering policy based on the full network topology and information of all nodes in the network.

For the distributed solution, few duplicate contents are cached at MENs. Specifically, content 16 is cached at MEN-1 and MEN-3, while content 17 is stored at MEN-2 and MEN-3. The reason is that the distributed solution cannot achieve optimal solution even though the duplicate con-
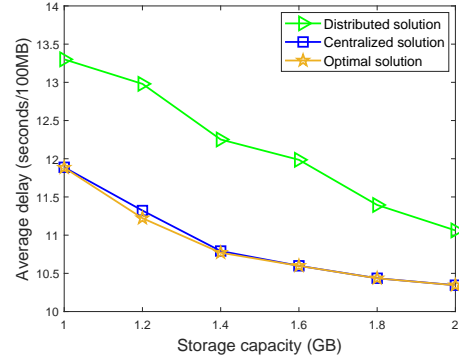

Fig. 4: Evaluation of proposed solutions against optimal solution.

tents at all MENs are minimized. However, compared with locally optimal policy, the distributed solution has much better disparate content distribution due to the horizontal cooperation (i.e., 17 of 30 available contents are cached in the network). When we use the locally optimal policy, more duplicate contents are produced (i.e., only 13 of 30 available contents are cached in the network) due to the locally caching optimization at each MEN (without horizontal cooperation among MENs).

7 SIMULATION RESULTS

We perform simulations to evaluate the performance of the proposed solutions, i.e., centralized and distributed solutions, with those of other caching policies including greedy policy [11], [20], most FoA policy [3], [33], locally optimal policy [6], [34], and guaranteed greedy policy [14], [35]. For the greedy policy, contents are cached as many as possible. For the most FoA policy, contents with high FoA are prioritized to be cached. For the locally optimal policy, the MENs minimize their own delays without considering the cooperation. For the guaranteed greedy policy, contents which maximize caching gain are greedily added into caching storage under guaranteed $(1 - 1/e)$ factor of the optimal solution. In all simulations, the content size is randomly generated using a uniform distribution between 100 and 300MB, while the FoAs of all MENs follow a Zipf distribution based on the ranks of the contents [10], [19] with shape parameter set at 0.1. The bandwidth between an MU and its associated MEN, and between two directly connected MENs are set at 10 and 45Mbps, respectively. Furthermore, bandwidth between an MEN and the BS is 10Mbps and between the BS and the CS is 60Mbps. Note that the bandwidth between two MENs is usually higher than that between an MEN and the BS because in practice two directly connected MENs are often placed in the same area or close to each other where wired or fast wireless connections can be used [10], [19].

7.1 Evaluation of Proposed vs. Optimal Solutions

Fig. 4 demonstrates the total average delay performance between our proposed solutions, i.e., centralized and distributed solutions, and optimal solution when the storage capacity increases from 1 to 2GB and 15 contents are
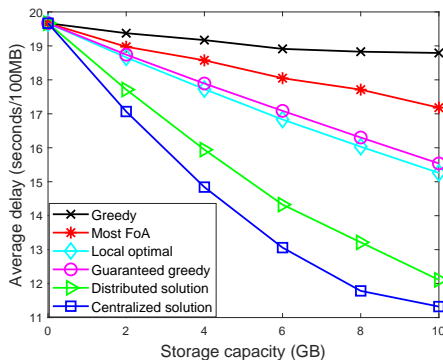
Fig. 5: Average delay vs storage capacity.

available for 2 MENs. Here, we only consider small-size problems due to the exponential complexity of finding the optimal solution using the exhaustive searching scheme. It can be seen that although the proposed centralized solution cannot guarantee the optimal solution for the original problem, its performance can achieve very close (less than 1% gap) to that of the optimal solution obtained by the exhaustive searching scheme. Furthermore, the proposed distributed solution still can provide reasonable total average delay gap within 13% to the optimal solution and the gap becomes smaller as the storage capacity increases.

7.2 Total Average Delay

In this section, we evaluate the total average delay under the influence of various storage capacities, number of contents, and number of MENs. In particular, we use 0 to 10GB caching capacity, 50 to 200 number of available contents, and 2 to 10 number of MENs. Moreover, the content size is uniformly distributed between 100MB and 300MB.

7.2.1 Effects of Storage Capacity

Fig. 5 shows the trend of the total average delay in the MEC network with 200 considered available contents as the storage capacity increases from 0 to 10GB. It is observed that as the storage capacity increases, the average delays obtained by all policies are reduced, and the centralized solution outperforms all other policies. Specifically, when the storage capacity is 10GB, the average delay obtained by the centralized solution is 35% lower than the locally optimal policy, 37% lower than the guaranteed greedy policy, 52% lower than the most FoA policy, and 66% lower than the greedy policy. This is because the MENs can collaborate together to improve caching efficiency for the whole network when the centralized solution is used. In constrast, conventional policies (i.e., greedy, most FoA, guaranteed greedy, and locally optimal policies) do not consider cooperative caching among the MENs, and thus requested contents can only be downloaded from the CS (through the BS) if the contents are not stored at the associated MEN of the requesting users. For the distributed solution, although it cannot achieve the optimal solution, its performance is always better than the conventional policies, and the gap with the centralized solution is only within 12%.

We then observe the average delay from each participating MEN and the BS in Fig. 6. In general, each MEN

in Fig. 6(a)-(d) follows the same trend as Fig. 5. However, an interesting result can be seen in Fig. 6(e). In our framework, we leverage the direct horizontal collaboration among MENs to minimize the average delay for the whole network. As such, since all nodes are also connected to the BS directly, the BS must sacrifice its performance to reduce the overall network delay by allowing other nodes to cache more popular contents. Nonetheless, even the BS sacrifices, its performance is still greater than those of the greedy and most FoA policies, and close to those of the locally optimal and guaranteed greedy policies.

7.2.2 Effects of Number of Contents

Fig. 7 presents the total average delay as the number of contents increases from 50 to 200 contents. We fix the storage capacity for all MENs (including the BS) at 5GB. As expected, for a given storage capacity, if the number of contents increases, the average delays obtained by all policies increase. The reason is that the mobile users cannot download more contents from their directly connected MENs. Instead, the MENs need to download the contents from other nodes more frequently. Nevertheless, due to the collaboration, the average delay obtained by the centralized solution is still much lower by 15% up to 55% than those of all other conventional policies. Furthermore, the distributed solution can achieve the performance very close to that of the centralized solution especially when the number of contents is low.

7.2.3 Effect of Number of MENs

Fig. 8 shows the total average delay when we increase the number of MENs from 2 to 10. We fix the storage capacity of all MENs and the number of contents at 10GB and 200 contents, respectively. Interestingly, as the number of MENs increases, the total average delays obtained by the greedy, most FoA, guaranteed greedy, and locally optimal policies increase gradually. The reason is that these methods do not consider the collaboration among the MENs, and thus when the number of MENs increases, the delay at each MEN will contribute to the total delay of the network, yielding to an increasing trend.

Nevertheless, the total average delay obtained by the distributed solution first dramatically decreases when the number of MENs increases from 2 to 4 (the same trend applies for the centralized solution). Then, the total average delay of the distributed solution slightly rises. However, it is still much lower by as much as 29% compared with other conventional policies as the number of MENs increases. The reason of increasing trend in the total delay is that each MEN does not have full knowledge of the content caching decisions in the whole network. Thus, the MEN may download the content from the CS (through the BS) eventhough not directly connected MENs may cache the requested content. In contrast, due to the centralized control, the centralized solution can slightly reduce more delay and remain stable when the number of MENs exceeds 8. In this case, the total average delay of the centralized solution is at least 40% lower than those of other conventional policies. Specifically, the centralized solution can utilize the efficiency of collaboration among the MENs, and thus as the number of MENs increases, more connections will be created,

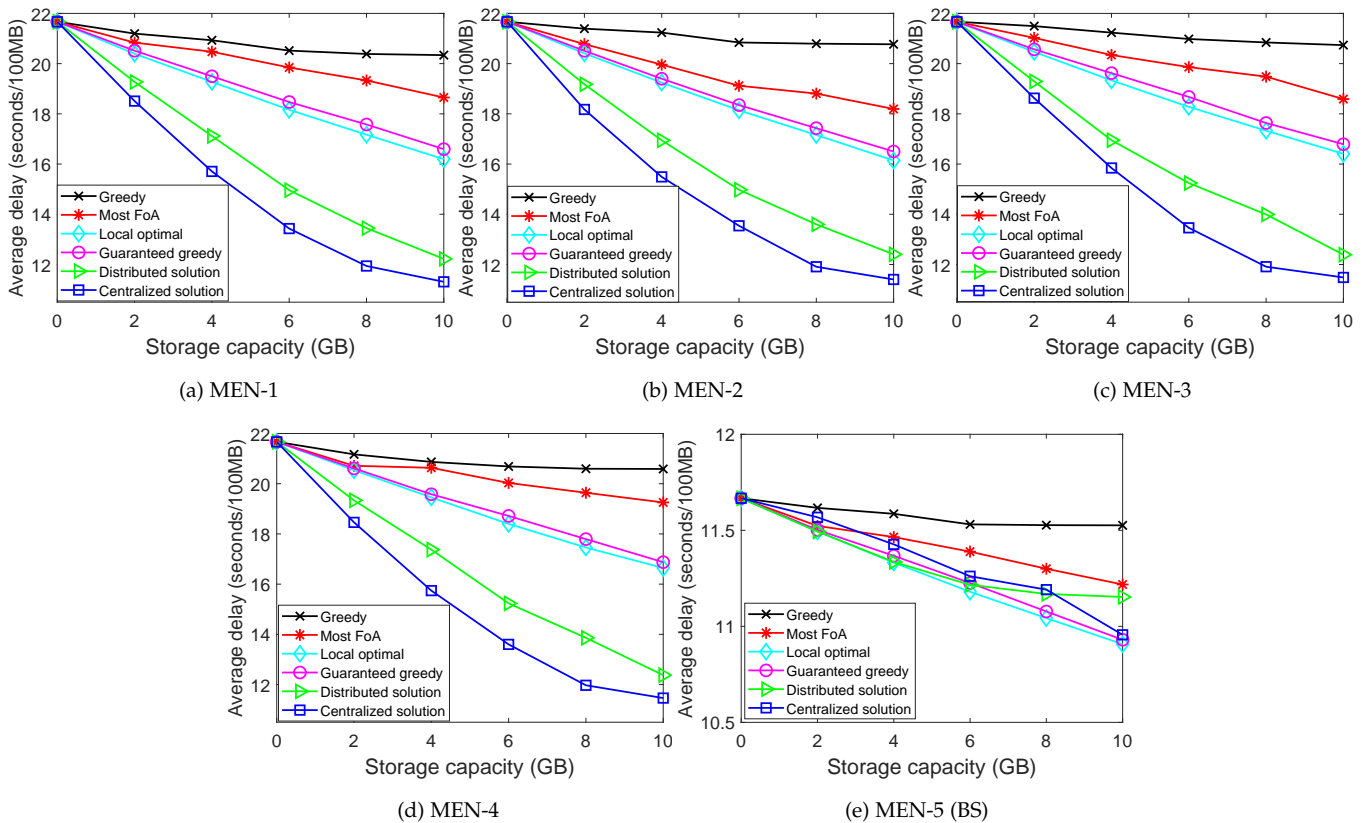(a) MEN-1 (b) MEN-2 (c) MEN-3

(d) MEN-4 (e) MEN-5 (BS)

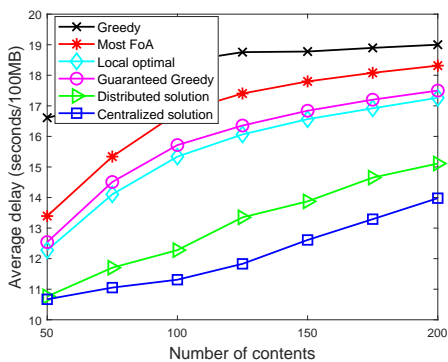Fig. 6: Average delay vs storage capacity for the MENs and the BS.

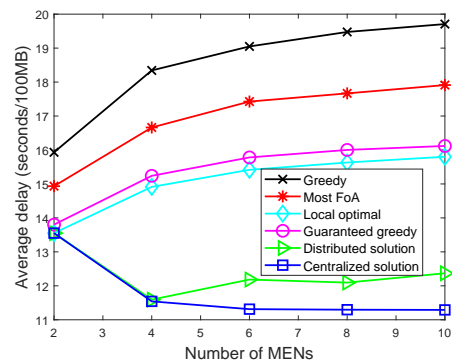Fig. 7: Average delay vs number of contents.

Fig. 8: Average delay vs number of MENs.

thereby better leveraging the collaboration among MENs. We also notice that the delay reduction of the centralized solution gets saturated after the number of MENs reaches a given number, indicating that all available contents are cached at MENs. Fig. 8 also provides useful information to help the MEC service providers effectively deploy MENs.

7.3 Cache Hit Ratio Probability

To further evaluate the performance of the proposed solutions, we conduct simulations to show the cache hit ratio of the caching policies. Specifically, we generate incoming content requests randomly using a uniform distribution. In addition, we consider two types of the cache hit ratio: (1) *local-hit* and (2) *global-hit*. The former represents the cache

hit ratio at each individual MEN, while the latter captures the cache hit ratio from the whole network. Specifically, we denote h_n as the local-hit of MEN-n. Then, the average cache hit ratio for the whole network of other caching policies (i.e., greedy, most FoA, guaranteed greedy, and locally optimal policies) h_{tot} is

$$h_{tot} \overset{\text{def}}{=} \frac{1}{N}\left[\sum_{n=1}^{N-1} h_n + N h_N\right], \qquad (54)$$

and of the proposed solutions h_{tot}^* is

$$h_{tot}^* \overset{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N}\left(h_n + \sum_{m\in\mathcal{M}_n^i, m\neq n} h_m\right), \forall i \in \mathcal{I}, \qquad (55)$$

where h_N and $\sum\limits_{m \in \mathcal{M}_n^i, m \neq n} h_m$ are the global-hit for the conventional policies and the proposed solutions, respectively.

Fig. 9 presents the average cache hit ratios for all scenarios. Note that, for the greedy policy we only show the local-hit because the greedy policy is based on the sizes of contents only, and thus the MENs and the BS have the same set of cached contents. As observed in Fig. 9, the average cache hit ratios of most FoA, guaranteed greedy, and locally optimal policies are greater than that of the greedy policy because they have less duplicate contents on the MENs. In particular, the MENs may have different users' demands, and thus they may cache dissimilar contents. For the distributed and centralized solutions, by leveraging the direct horizontal collaboration, their cache hit ratios (including local-hit and global-hit) are up to 4 times greater than those of other policies. These figures also clearly show impacts of the collaboration among MENs. In particular, although the local-hit may not always be the best (because MENs may sacrifice to cache contents for other nodes to minimize the average delay for the whole network), the total cache hit ratio for the network obtained by the centralized solution always achieves the highest value. It is also worth noting that when the total cache hit ratio in the network increases, the traffic load on the backhaul network will be reduced.

8 CONCLUSION

In this paper, we have introduced the effective joint cooperate caching and delivering framework (JOCAD) by leveraging direct horizontal cooperation among MENs. This framework aims to minimize the total average delay for the MEC network and lessen the network traffic on the backhaul network. To address the optimal joint caching and delivering problem, we have proposed the novel transformation method together with the improved branch-and-bound algorithm with the interior-point method. To reduce the complexity and communication overheads among MENs, we have also introduce distributed cooperative caching-delivering solution which minimizes the duplicate contents among directly connected MENs. Through the simulation results, we have shown that the proposed solutions can significantly outperform other caching policies in terms of the total average delay and cache hit ratio. Furthermore, the results can provide useful knowledge for MEC service providers to tradeoff between the quality of service and the implementation costs in the MEC network. To further guarantee the performance, in the future work, we will develop effective distributed algorithms which can theoretically show the bounds and compare their performance with the current proposed solutions.

REFERENCES

[1] Cisco Mobile Visual Networking Index. Available Online: https://newsroom.cisco.com/press-release-content?articleId=1819296. Last Accessed on June 2018.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322-2358, Fourth Quarter 2017.

[3] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 131-145, Jan. 2016.

[4] K. Li, M. Tao, and Z. Chen, "Exploiting computation replication in multi-user multi-server mobile edge computing networks," in *IEEE GLOBECOM*, Dec. 2018, pp. 1-7.

[5] K. Li, C. Yang, Z. Chen, and M. Tao, "Optimization and analysis of probabilistic caching in N-tier heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1283-1297, Feb. 2018.

[6] D. T. Hoang, D. Niyato, D. N. Nguyen, E. Dutkiewicz, P. Wang and Z. Han, "A dynamic edge caching framework for mobile 5G networks," *IEEE Wireless Commun.*, vol. 25, no. 5, pp. 95-103, Oct. 2018.

[7] 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, Fourth Quarter 2016.

[8] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *IEEE INFOCOM 2016*, Apr. 2016, pp. 1-9.

[9] J. Wen, K. Huang, S. Yang, and V. O. K. Li, "Cache-enabled heterogeneous cellular networks: optimal tier-level content placement," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5939-5952, Sept. 2017.

[10] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," in *IEEE INFOCOM*, Apr. 2016, pp. 1-9.

[11] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *IEEE INFOCOM*, Mar. 2010, pp. 1-9.

[12] 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.

[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] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *IEEE INFOCOM*, Apr. 2015, pp. 936-944.

[15] S. Kekki, *et al.*, "MEC in 5G networks," *ETSI White Paper*, no. 28, Jun. 2018.

[16] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212-3229, Jun. 2016.

[17] Q. Li, W. Shi, X. Ge, and Z. Niu, "Cooperative edge caching in software-defined hyper-cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2596-2605, Nov. 2017.

[18] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995-3007, Apr. 2016.

[19] W. C. Ao and K. Psounis, "Distributed caching and small cell cooperation for fast content delivery," in *ACM MOBIHOC*, Jun. 2015, pp. 127-136.

[20] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. S. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791-1805, Aug. 2018.

[21] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: from classical to evolutionary approaches and applications," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 276-295, Apr. 2018.

[22] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter, "Branching and bounds tightening techniques for non-convex MINLP," *Optim. Method. Softw.*, vol. 24, no. 4-5, pp. 597-634, Aug. 2009.

[23] IBM ILOG CPLEX Optimization Studio CPLEX Users Manual version 12.7, IBM, 2016.

[24] *GAMS The Solver Manuals*, GAMS Development Corp, 2009.

[25] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373-395, Nov. 1984.

[26] R. Byrd, N. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM J. Optimiz.*, vol. 9, no. 4, pp. 877-900, Sept. 1999.
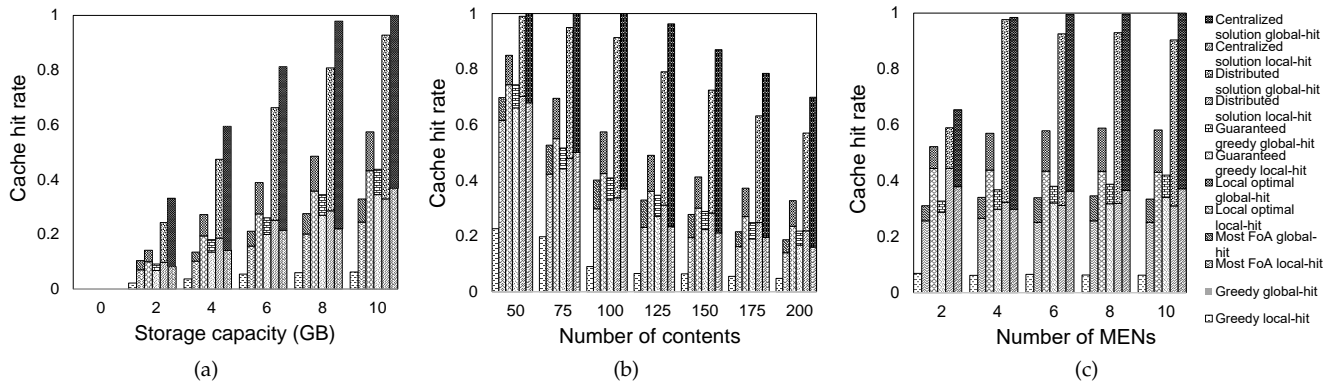
Fig. 9: Average cache hit ratio when (a) the storage capacity increases, (b) the number of contents increases, and (c) the number of MENs increases.

[27] A. Wachter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25-57, Mar. 2006.

[28] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: an integrated package for nonlinear optimization," *Large-Scale Nonlinear Optimization*, pp. 35-59, Springer, Boston, MA, 2006.

[29] H. Y. Benson, "Mixed integer nonlinear programming using interior-point methods," *Optim. Method. Softw.*, vol. 26, no. 6, pp. 911-931, Dec. 2011.

[30] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM J. Numer. Anal.*, vol. 20, no. 3, pp. 626-637, Jun. 1983.

[31] D. R. Smith, "On the computational complexity of branch and bound search strategies," Nov. 1979.

[32] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, pp. 85-103, 1972.

[33] Y. Cui and D. Jiang, "Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 250-264, Jan. 2017.

[34] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in fog-RANs: from centralized to distributed algorithms," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7039-7051, Nov. 2017.

[35] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: wireless video content delivery through distributed caching helpers," in *IEEE INFOCOM*, Mar. 2012, pp. 1107-1115.

[36] R. M. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, no. 1, pp. 45-68, Jan. 1975.

[37] D. R. Smith, "Random trees and the analysis of branch and bound procedures," *J. ACM*, vol. 31, no. 1, , pp. 163-188, Jan. 1984.

[38] J. M. Fowkes, N. I. M. Gould, and C. L. Farmer, "A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions," *J. Global Optim.*, vol. 56, no. 4, pp. 1791-1815, Aug. 2013.

Dinh Thai Hoang (M'16) is currently a faculty member at the School of Electrical and Data Engineering, University of Technology Sydney, Australia. He received his Ph.D. in Computer Science and Engineering from the Nanyang Technological University, Singapore, in 2016. His research interests include emerging topics in wireless communications and networking such as ambient backscatter communications, vehicular communications, cybersecurity, IoT, and 5G networks. He is an Exemplary Reviewer of IEEE Transactions on Communications in 2018 and an Exemplary Reviewer of IEEE Transactions on Wireless Communications in 2017 and 2018. Currently, he is an Editor of IEEE Wireless Communications Letters and IEEE Transactions on Cognitive Communications and Networking.

Diep N. Nguyen is a faculty member of the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS). He received M.E. and Ph.D. in Electrical and Computer Engineering from the University of California San Diego (UCSD) and The University of Arizona (UA), respectively. Before joining UTS, he was a DECRA Research Fellow at Macquarie University, a member of technical staff at Broadcom (California), ARCON Corporation (Boston), consulting the Federal Administration of Aviation on turning detection of UAVs and aircraft, US Air Force Research Lab on anti-jamming. He has received several awards from LG Electronics, University of California, San Diego, The University of Arizona, US National Science Foundation, Australian Research Council. His recent research interests are in the areas of computer networking, wireless communications, and machine learning application, with emphasis on systems performance and security/privacy.

Yuris Mulya Saputra (SM'18) received his B.E. degree in Telecommunication Engineering from the Institut Teknologi Bandung (ITB), Indonesia in 2010, and his M.Sc degree in Electrical and Information Engineering from the Seoul National University of Science and Technology (SeoulTech), South Korea in 2014. He is currently a Ph.D. student at the University of Technology Sydney (UTS), Australia and a full-time lecturer in Universitas Gadjah Mada (UGM), Indonesia. Before joining UTS and UGM, he was a researcher in the Intelligent Systems Research Group, SeoulTech, South Korea, from 2014 to 2015 and an application software developer in the Digital Appliance Division, Samsung Electronics, Indonesia, from 2010 to 2012. His research interests include mobile computing, energy efficiency, and optimization problems for wireless communication networks.

Eryk Dutkiewicz received his B.E. degree in Electrical and Electronic Engineering from the University of Adelaide in 1988, his M.Sc. degree in Applied Mathematics from the University of Adelaide in 1992 and his PhD in Telecommunications from the University of Wollongong in 1996. His industry experience includes management of the Wireless Research Laboratory at Motorola in early 2000s. Prof. Dutkiewicz is currently the Head of School of Electrical and Data Engineering at the University of Technology Sydney, Australia. He is a Senior Member of IEEE. He also holds a professorial appointment at Hokkaido University in Japan. His current research interests cover 5G and IoT networks.

APPENDIX A
PROOF OF LEMMA 1

To prove this Lemma, we prove that one special case of the problem is NP-Complete. Specifically, in the problem $(\mathbf{P}_1)$, let's not take into account the delivering problem. In particular, if MEN-n does not have the requested content, it will download the content directly from the CS via the BS. Then, the problem $(\mathbf{P}_1)$ can be formulated as follows:

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(x_n^i d_\alpha + (1-x_n^i)d_\delta \right) \right]. \tag{56}$$

Based on Eq. (56), the problem becomes binary linear programming (with $I \times N$ number of binary variables) which was proved to be NP-complete in [32] and [36]. Since the special case problem is at least as hard as the hardest problems in NP (i.e., NP-complete problem), the nested dual binary nonlinear programming problem $(\mathbf{P}_1)$ is NP-hard.

APPENDIX B
PROOF OF THEOREM 1

In the following, we will prove that if $\mathbf{x}^*$ is the optimal solution of $(\mathbf{P}_1)$, then it is also the optimal solution of $(\mathbf{P}_2)$ and vice versa. We first recall the inner minimization $F_{IL}(\mathbf{x})$ in $(\mathbf{P}_1)$ as follows:

$$\min_{m} \left([x_m^i + (1-x_m^i)V] \frac{c_i}{l_n^m} \right) \overset{\text{def}}{=} \\ \min_{m} \left(Q_n^i(x_1^i), \dots, Q_n^i(x_m^i), \dots, Q_n^i(x_{M_n^i}^i) \right), \tag{57}$$

where $Q_n^i(x_m^i) \overset{\text{def}}{=} [x_m^i + (1-x_m^i)V] \frac{c_i}{l_n^m}$ and M_n^i is the total number of directly connected MENs containing content i for MEN-n, with $\forall m \in \mathcal{M}_n^i \subset \mathcal{N}, \forall n \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$. Then we can rewrite the objective function in Eq. (8) into

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}) + \right. \right. \\ \left. \left. \Phi(\mathbf{x}) \min_{m} \left(Q_n^i(x_1^i), \dots, Q_n^i(x_m^i), \dots, Q_n^i(x_{M_n^i}^i) \right) \right) \right], \tag{58}$$

where $\mathbf{x} = (x_n^i, x_m^i)$ with $\forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$. Suppose that $\mathbf{x}^* = (\hat{x}_n^i, \hat{x}_m^i), \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$ is the optimal solution of $F(\mathbf{x})$ in Eq. (58). Without loss of generality, $m = k_n^i$ will be selected as the MEN which represents the minimum delivering decision to download content i for MEN-n if the following condition satisfies:

$$Q_n^i(\hat{x}_{m=k_n^i}^i) \leq Q_n^i(\hat{x}_{m \neq k_n^i}^i), \\ \forall n, m, k_n^i \in \mathcal{N}, m \neq n, k_n^i \neq n, \forall i \in \mathcal{I}. \tag{59}$$

Hence, $\min_{\mathbf{x}} F(\mathbf{x})$ can achieve the optimal objective function as follows:

$$F(\mathbf{x}^*) = \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) Q_n^i(\hat{x}_{m=k_n^i}^i) \right) \right]. \tag{60}$$

Next, from MINLP problem $(\mathbf{P}_2)$, we can also obtain the following expression:

$$\min_{\{\mathbf{x},\mathbf{y},\mathbf{z}\}} F(\mathbf{x},\mathbf{y},\mathbf{z}) = \\ \min_{\{\mathbf{x},\mathbf{y},\mathbf{z}\}} \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}) + \Phi(\mathbf{x}) z_n^i \right) \right], \tag{61} \\ \text{s.t. (13)-(16).}$$

Given the optimal solution $\mathbf{x}^* = (\hat{x}_n^i, \hat{x}_m^i), \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$ from the Eq. (58), we can rewrite the Eq. (61) by:

$$\min_{\{\mathbf{x}^*,\mathbf{y},\mathbf{z}\}} F(\mathbf{x}^*,\mathbf{y},\mathbf{z}) = \\ \min_{\{\mathbf{x}^*,\mathbf{y},\mathbf{z}\}} \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) z_n^i \right) \right], \\ \text{s.t.} \begin{cases} z_n^i \geq Q_n^i(\hat{x}_m^i) - V y_m^i, \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}, \\ \sum_{m \in \mathcal{M}_n^i} y_m^i = M_n^i - 1, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \\ y_m^i \in \{0,1\}, \forall m \in \mathcal{N}, \forall i \in \mathcal{I}, \\ z_n^i \in \mathbb{R}_0^+, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}. \end{cases} \tag{62}$$

Then, consider all feasible values of y_m^i in Eq. (62). Based on the constraints $\sum_{m \in \mathcal{M}_n^i} y_m^i = M_n^i - 1$, there are one MEN-$m$ with $y_m^i = 0$ and $M_n^i - 1$ MENs with $y_m^i = 1$. Suppose that $m = \kappa_n^i$, where $\forall \kappa_n^i \in \mathcal{M}_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$, is the current MEN which has $y_{m=\kappa_n^i}^i = 0$. Thus, $z_n^i \geq Q_n^i(\hat{x}_{m=\kappa_n^i}^i)$ for $m = \kappa_n^i$ and $z_n^i \geq Q_n^i(\hat{x}_{m \neq \kappa_n^i}^i) - V$ for $m \neq \kappa_n^i$. Since V is a very big value, we can eliminate the rest of the constraints when $m \neq \kappa_n^i$. As a result, only $z_n^i \geq Q_n^i(\hat{x}_{m=\kappa_n^i}^i), \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$ is applied for each possible optimization as follows:

$$\min_{\{\mathbf{x}^*,\mathbf{z}\}} F(\mathbf{x}^*,\mathbf{z}) = \\ \min_{\{\mathbf{x}^*,\mathbf{z}\}} \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) z_n^i \right) \right], \tag{63} \\ \text{s.t.} \quad z_n^i \geq Q_n^i(\hat{x}_{m=\kappa_n^i}^i), \forall n \in \mathcal{N}, \forall i \in \mathcal{I}.$$

If $\mathbf{z}^* = \hat{z}_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$ is also the optimal solution, then the possible optimal objective value $F(\mathbf{x}^*, \mathbf{z}^*)$ for each κ_n^i is

$$F(\mathbf{x}^*, \mathbf{z}^*) = \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) \hat{z}_n^i \right) \right], \tag{64}$$

where $\hat{z}_n^i = Q_n^i(\hat{x}_{m=\kappa_n^i}^i), \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$. Therefore, we can derive:

$$F(\mathbf{x}^*) = \sum_{i=1}^{I} \sum_{n=1}^{N} f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) Q_n^i(\hat{x}_{m=\kappa_n^i}^i) \right) \right]. \tag{65}$$

Based on Eq. (59), from each MEN-n we can choose an MEN-m to download content i with minimum delivering time, i.e., $Q_n^i(\hat{x}_{m=k_n^i}^i) \leq Q_n^i(\hat{x}_{m \neq k_n^i}^i)$. In other words, the selected node κ_n^i in $(\mathbf{P}_2)$ is the same as the selected node k_n^i in $(\mathbf{P}_1)$, $\forall n \in \mathcal{N}$ and $\forall i \in \mathcal{I}$. As a result, if $\mathbf{x}^*$ is the optimal solution of $(\mathbf{P}_1)$, it is also the optimal solution of $(\mathbf{P}_2)$.

Similarly, if $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is the optimal solution of $(\mathbf{P}_2)$, we can prove that $\mathbf{x}^*$ is also the optimal solution of $(\mathbf{P}_1)$. We recall MINLP problem $(\mathbf{P}_2)$ in Eq. (61). Given that $\mathbf{x}^* = (\hat{x}_n^i, \hat{x}_m^i)$, $\mathbf{y}^* = \hat{y}_m^i$, and $\mathbf{z}^* = \hat{z}_n^i, \forall n, m \in \mathcal{N}, m \neq n, \forall i \in \mathcal{I}$

are the optimal solutions of $(\mathbf{P}_2)$ such that

$$\hat{z}_n^i = Q_n^i(\hat{x}_{m=\kappa_n^i}^i), \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{66}$$

where $m = \kappa_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$ is the current MEN which has $\hat{y}_{m=\kappa_n^i}^i = 0$. After ignoring other MENs when $m \neq \kappa_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$ because of $\hat{y}_{m \neq \kappa_n^i}^i = 1$ and the influence of big value V, then we have

$$F(\mathbf{x}^*, \mathbf{z}^*) = \sum_{i=1}^I \sum_{n=1}^N f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) \hat{z}_n^i \right) \right], \tag{67}$$

where $\hat{z}_n^i = Q_n^i(\hat{x}_{m=\kappa_n^i}^i), \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$, and thus

$$F(\mathbf{x}^*) = \sum_{i=1}^I \sum_{n=1}^N f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) Q_n^i(\hat{x}_{m=\kappa_n^i}^i) \right) \right]. \tag{68}$$

From $(\mathbf{P}_1)$, we also have $m = k_n^i$, where $\forall k_n^i \in \mathcal{M}_n^i, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}$, as the selected MEN which indicates the minimum delivering decision to download content i for MEN-n if

$$Q_n^i(\hat{x}_{m=k_n^i}^i) \leq Q_n^i(\hat{x}_{m \neq k_n^i}^i), \\ \forall n, m, k_n^i \in \mathcal{N}, m \neq n, k_n^i \neq n, \forall i \in \mathcal{I}. \tag{69}$$

Considering the condition in Eq. (66) and Eq. (69), there exists MENs such that $m = \kappa_n^i = k_n^i, \forall n \in \mathcal{N}$ and $\forall i \in \mathcal{I}$. Hence, without loss of generality, we can achieve the final optimal objective value of $(\mathbf{P}_1)$ as follows:

$$F(\mathbf{x}^*) = \sum_{i=1}^I \sum_{n=1}^N f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) Q_n^i(\hat{x}_{m=\kappa_n^i}^i) \right) \right] \\ = \sum_{i=1}^I \sum_{n=1}^N f_n^i \left[\sum_{u=1}^{U_n} \left(\Omega(\mathbf{x}^*) + \Phi(\mathbf{x}^*) Q_n^i(\hat{x}_{m=k_n^i}^i) \right) \right], \tag{70}$$

which is the same as $F(\mathbf{x}^*, \mathbf{z}^*)$ in Eq. (67). Thus, we can conclude that $(\mathbf{P}_1)$ is equivalent to $(\mathbf{P}_2)$.

APPENDIX C
PROOF OF THEOREM 2

We adopt this proof from [31] and [37]. Suppose that T_ρ specifies the expected number of total average delays searched in the depth ρ. Then, we obtain that $T_0 = 1$ at root problem $(\mathbf{RP})$ and $T_\rho \geq 1$ for $\rho \geq 1$, i.e., at subproblems $(\mathbf{SP}s)$. Thus, we can derive T by

$$T = 1 + \sum_{\rho=1}^\varrho \geq 1 + \sum_{\rho=1}^\varrho 1 \geq 1 + \varrho. \tag{71}$$

Furthermore, there exists a constant ς such that $T_\varrho \leq \varsigma \leq T_\infty$ for all ϱ, where $T_\infty = \lim_{\varrho \to \infty} T_\varrho$, and $T_0 \leq T_1 \leq \ldots \leq T_\varrho \leq \varsigma$. Then, we can express that

$$T \leq 1 + \sum_{\rho=1}^\varrho \varsigma \leq 1 + \varsigma \varrho. \tag{72}$$

From Eq. (71) and Eq. (72), it implies that $1 + \varrho \leq T \leq 1 + \varsigma \varrho$. Alternatively, we can state that T is linear (or polynomial) at the depth ϱ.

APPENDIX D
PROOF OF THEOREM 3

We adopt this proof from [38]. Given that the t-th total average delay $\Psi_t = \arg \min_t F_{\mathbf{SP}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and ψ_t is a search

region to obtain Ψ_t for some $\theta_t < t$. Then, there exists $\phi > 0$ such that for $\eta > 0$ and any $\mathbf{x}, \mathbf{y}, \mathbf{z}$, we have

$$F_{\mathbf{SP}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \leq \phi \implies \beta_U(\Psi_t) - \beta_L(\Psi_t) \leq \eta. \tag{73}$$

Then, ψ_t should also have $F_{\mathbf{SP}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \leq \phi$ when $t = \hat{t} \in \mathbb{N}$. Based on Eq. (73), we can obtain

$$\beta_U(\psi_{\hat{t}}) - \beta_L(\psi_{\hat{t}}) \leq \eta. \tag{74}$$

Since $\psi_{\hat{t}}$ is split (to create two new subproblems) at step $\theta_{\hat{t}}$, then $\beta_L(\psi_{\hat{t}}) = \beta_L^{(\theta_{\hat{t}})}$. As a result, the condition becomes

$$\beta_U^{(\theta_{\hat{t}})} - \beta_L^{(\theta_{\hat{t}})} \leq \beta_U(\psi_{\hat{t}}) - \beta_L^{(\theta_{\hat{t}})} \leq \eta, \tag{75}$$

and thus $\beta_U^{(\theta_{\hat{t}})} \leq \beta_U(\psi_{\hat{t}})$. Next, consider that the final total average delay $\hat{\Psi}$ is obtained at $\mathbf{x}^*$, $\mathbf{y}^*$, and $\mathbf{z}^*$. To satisfy a condition that $\beta_L^{(\theta_{\hat{t}})} \leq \hat{\Psi}$, the following expression can be obtained based on Eq. (75):

$$\beta_U^{(\theta_{\hat{t}})} - \hat{\Psi} \leq \beta_U^{(\theta_{\hat{t}})} - \beta_L^{(\theta_{\hat{t}})} \leq \eta, \tag{76}$$

where $\beta_U^{(\theta_{\hat{t}})}$ is shown within the optimality tolerance η of the $F(\mathbf{x}, \mathbf{y}, \mathbf{z})$.