Multidimensional Cooperative Caching in CoMP-Integrated Ultra-Dense Cellular Networks

Peng Lin, *Member, IEEE*, Qingyang Song, *Senior Member, IEEE*, and Abbas Jamalipour, *Fellow, IEEE*

***Abstract*— Small base stations (BSs) equipped with caching units are potential to provide improved quality of service (QoS) for multimedia service in ultra-dense cellular networks (UDCNs). In addition, the Coordinated MultiPoint (CoMP) transmission method, allowing multiple BSs to jointly serve users, is proposed to increase the throughput of cell-edge mobile terminals (MTs). Yet, the combination of content caching and CoMP in UDCNs is still not well explored for future networks. In this paper, we focus on the application of caching in CoMP-integrated UDCNs, where the cache-enabled BSs can collaboratively serve each MT using either joint transmission or single transmission. We propose a multidimensional cooperative caching (MDCC) scheme, supporting storage-dimension and transmission-dimension cooperations for the content placement. In particular, we analyze the delivery delay based on request patterns, transmission method, and the proposed cooperation strategy. Then the content placement problem is formulated as a problem of minimizing the overall expected delay. The problem is a mixed binary integer linear programming (BILP) problem, which is NP-hard. Therefore, we address the problem with approximation and substitution, and design a genetic algorithm (GA) based method to solve it. Simulation results demonstrate that the proposed MDCC scheme contributes performance gain in terms of content delivery delay in both cell-core and cell-edge areas.**

***Index Terms*— Cooperative caching, CoMP, ultra-dense cellular networks, joint transmission, genetic algorithm.**

I. Introduction

THE world-wide growth of smart mobile terminals (MTs) has triggered an explosion of mobile Internet traffic. Especially, multimedia transmission has become a burden to radio access networks (RANs) and backhaul networks [1]–[3]. However, according to [2] released by Cisco, it is observed that the increased traffic is mostly caused by duplicated downloading of popular contents. To support the tremendous growth of mobile video/audio streaming without bringing the associated problems of congestion, delay, and lack of capacity, researchers have proposed revolutionary schemes for the deployment of next-generation mobile networks [4].

An emerging technique for reducing the backhaul traffic is to cache popular contents locally at intermediate nodes, such as cellular base stations (BSs), during off-peak times [5]–[16]. In this way, the files requested by MTs can be transmitted directly from the caches in RANs, instead of having to be fetched from the original server. Two deterministic caching schemes of most popular content (MPC) and largest content diversity (LCD) are commonly used to offload wireless cellular traffic [6], [7]. The MPC scheme can provide optimal performance in nonoverlapping cells. Correspondingly, in overlapping coverage areas, the LCD scheme can contribute high hit rate by increasing the diversity of cached contents. The tradeoff between the MPC and LCD schemes is investigated by [8], where a hybrid caching placement scheme that enables two degrees of flexibility to combine MPC and LCD is proposed. In addition to the deterministic caching schemes, some probabilistic placement policies [9]–[11] are introduced to increase cache hit rate, but they are still isolated caching without considering the cooperation between BSs.

To further improve cache hit rate and make efficient utilization of the limited storage resource, several schemes have been proposed to explore cooperative caching from different perspectives. For instance, in a conventional cooperation-based caching scheme [12], the caching information of distributed caching units are shared with broadcast messages and content placement is optimized for homogenous content demands. Reference [13] presents a collaborative caching framework for a three-tier heterogeneous network, where the cached files are shared among macro, micro, and pico BSs, and an efficient multitier optimization of content caching is proposed to offload traffic. The degree of cooperation between caching nodes is extended by [14], where three types of cooperation (including inter-BS, inter-device, and cross-tier) are considered when deploying content placement. Some other works like FemtoCaching [15] and CBCS-Caching [16] consider the scenario where adjacent cells overlap with each other. The corresponding BSs cooperatively cache contents so that they are regarded as an ensemble to serve MTs. Note that all these studies propose cooperative caching scheme under the consideration of traditional single transmission (ST) scenarios,

Manuscript received September 8, 2018; revised June 3, 2019 and November 5, 2019; accepted December 9, 2019. Date of publication December 24, 2019; date of current version March 10, 2020. This work was supported in part by the National Nature Science Foundation of China under Grant 61775033 and Grant 61771120 and in part by the Fundamental Research Funds for the Central Universities under Grant N171602002. The associate editor coordinating the review of this article and approving it for publication was A. Zaidi. *(Corresponding author: Qingyang Song.)*

P. Lin is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: penglin11@foxmail.com).

Q. Song is with the School of Information and Communication Engineering, Chongqing University of Post and Telecommunications, Chongqing 400065, China, and also with School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: songqy@cqupt.edu.cn).

A. Jamalipour is with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW 2006, Australia (e-mail: a.jamalipour@ieee.org).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2960329

in which each MT can only be served by a single BS at any given same time. However, 5G networks will support diverse transmission techniques, such as joint transmission (JT). When more than one BS can serve a single MT simultaneously by performing JT in an ultra-dense cellular network (UDCN), the optimal content placement scheme is expected to be different from the ST case. Therefore, the benefits of cooperative caching will depend not only on what content should be cached but also on how the cached content is transmitted.

If a file requested by a MT is cached in several nearby BSs, a multi-input single-output (MISO) channel is formed between the BSs and the MT and the BSs can collaboratively transmit the file to the MT by JT. This technique is referred to as Coordinated MultiPoint Joint Transmission (CoMP-JT), and has been standardized in LTE-Advanced systems [17], [18]. It helps reduce inter-cell interference and increase cell-edge throughput. Therefore, instead of only taking content diversity and content popularity into account in the process of ST-oriented content placement, transmission method selection is an additional consideration in the design of caching strategies for the CoMP-integrated system, where both ST and JT are available.

There are some attempts to combine content placement with content transmission. A cache-enabled opportunistic CoMP framework for wireless video streaming is originally proposed in [19]. In this framework, the relays and BSs cache a portion of video files, and then opportunistically employ CoMP to achieve MIMO cooperation gain and reduce the backhaul load. A popularity-based combinational caching strategy is proposed by [20], where cache space is partitioned to store the most popular contents and less-popular contents. By balancing transmission diversity and content diversity, cache serving probability and energy efficiency are optimized. Based on [20], the relationship between request successful probability and average transmission outage is investigated by [21]. In this work, a caching strategy is designed for popular file selection and placement. With this strategy, average outage probability can be reduced by adjusting the parameters of deciding popular file placement. Some other studies, such as [22] and [23], focus on reducing content delivery delay in the coexistence of ST and JT. In order to facilitate more CoMP-JT opportunities, some very popular contents are copied in all the accessible BSs for cell-edge MTs. As a result, the cell-edge throughput is improved by JT.

However, the existing transmission-oriented cooperative caching strategies assign high priority to either ST or JT. On one hand, the ST-prioritized cooperative caching strategies fail to provide the MTs with higher transmit rate contributed by JT. On the other hand, the JT-prioritized cooperative caching strategies sacrifice storage space and cause high cache miss for ST-employed MTs. Therefore, in this paper, we study the content placement to make good use of both ST and JT. The motivations behind our work are based on the following observations.

1) In the CoMP-integrated network, to provide cell-edge MTs with low-delay services, a high-popularity content should be stimulated to be copied at multiple BSs to guarantee quantitative JTs. To further save the content delivery delay for the ST-employed MTs, content sharing between BSs should be encouraged with the consideration of BSs' handy fetching the requested contents from their neighboring BSs.
2) The requirements for JT and ST are related to the MTs' requests and their locations. To meet MTs' demands for JT and ST in the network, it is necessary to explore an appropriate content placement that can support plenty of JTs while ensuring a high cache hit rate for ST based on the MTs' request patterns.

Therefore, an integrated scheme that jointly optimizes content placement for JT and ST has the potential to significantly improve the system performance. These facts motivated us to exploit both storage-dimension and transmission-dimension cooperation in solving the optimal caching problem in the CoMP-integrated UDCN. The aim of our work is to minimize content delivery delay. Therefore, main contributions of this paper are as the followings:

1) We propose a multidimensional cooperative caching (MDCC) strategy which includes storage-dimension and transmission-dimension cooperation. The storage-dimension cooperation enables adjacent BSs to share contents within one-hop transmissions. The transmission-dimension cooperation helps BSs to create JT opportunities. To our knowledge, this is the first study on joint optimizations of storage-dimension content sharing, transmission-dimension content placement, and transmitting method selection in CoMP-integrated networks.
2) Under the proposed caching strategy, we formulate an optimization problem to minimize the content delivery delay. One important feature that distinguishes our work from others is that the content placement is optimized together with routing and transmitting decisions under the transmission and storage constraints. Therefore, the problem here is more challenging to solve and belongs to a mixed binary integer linear programming (BILP) problem, which is NP-hard in nature.
3) Further, to reduce complexity, we transform the optimization problem into a classic knapsack problem with approximation and substitution. Afterwards, a heuristic search method based on the Genetic Algorithm (GA) is proposed to solve the problem efficiently, and an updating and routing scheme is designed for the cached files.
4) Finally, we compare the proposed MDCC strategy with the FemtoMPC strategy (an extended strategy from FemtoCaching [15]), as well as an "*advanced*" FemtoMPC strategy. The results show that our strategy can always hold lower content delivery delay for MTs in both cell-core and cell-edge areas.

The remainder of this paper is organized as follows. Section II presents the network model and the multidimensional cooperative caching strategy. In Section III, we formulate the joint optimization problem to be a BILP problem. Then, in Section IV, we approximate the optimization problem and solve it with a GA-based heuristic method, and propose a file updating and routing scheme. The simulation results and analysis are given in Section V. Finally, we give our

TABLE I
NOTATIONS

Symbol	**definition**
$\mathcal{B}$	The set of base stations.
$\mathcal{N}_{b_i}$	The set of neighboring BSs of BS b_i.
$\mathcal{F}$	The set of files in the network.
$\mathcal{A}_{m_j}$	The set of BSs that perform JT for MT m_j.
P_{b_i,f_k}	Content popularity of file f_k at BS b_i.
λ_{b_i,m_j}	Request rate of MT m_j at BS b_i.
$\lambda_{b_i,m_j}^{f_k}$	Request rate of MT m_j for file f_k at BS b_i.
F	Number of files in the network.
B	Number of BSs in the network.
S_{f_k}	Size of file f_k.
C_{b_i}	Storage space of BS b_i.
x_{b_i,f_k}	Caching decision of BS b_i for file f_k.
$y_{b_i,b_r}^{f_k}$	Routing decision of BSs b_r to b_i for file f_k.
z_{b_i,f_k}	JT decision of BS b_i for file f_k.
r_{b_i,m_j}^{ST}	Transmit rate from BS b_i to MT m_j using ST.
$r_{\mathcal{A},m_j}^{\text{JT}}$	Transmit rate of using JT to MT m_j.
$r_{b_i}^{\text{CP}}$	Average transmit rate from CP to b_i.
r_{b_i,b_r}^{NE}	Average transmit rate from BS b_r to b_i.
$\mathcal{M}_{b_i}$	The set of MTs in the network.
$\mathcal{M}_{b_i}^{ed}$	The set of cell-edge MTs associated with b_i.
$\mathcal{M}_{b_i}^{co}$	The set of cell-core MTs associated with b_i.
$d_{b_i,m_j,f_k}^{\text{ST}}$	Transmit delay of f_k from b_i to m_j using ST.
$d_{\mathcal{A},m_j,f_k}^{\text{JT}}$	Transmit delay of f_k from $\mathcal{A}$ to m_j using JT.
$d_{b_i,b_r,f_k}^{\text{NE}}$	Delivery delay of file f_k from BS b_i to b_r.
d_{b_i,f_k}^{CP}	Delivery delay of file f_k from CP to bs b_i.
g_{b_i,m_j}	Instantaneous SINR from BS b_i to MT m_j.
τ	The SINR threshold for a cell-core MT.
$N(C_{b_i})$	Number of files can be cached by BS b_i.
N_{pop}	Number of individuals in a population.
N_{ele}	Number of selected elite individuals.
N_{gen}	Number of generations.
ε	Terminate threshold for population evolution.

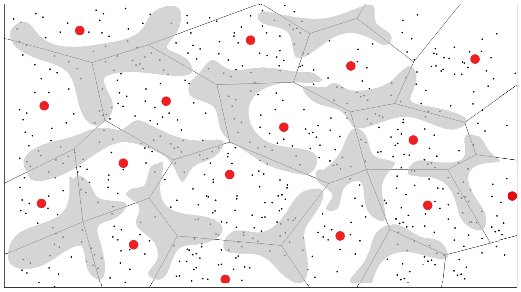


Fig. 1. Base stations (red circle) and mobile terminals (black spot) in ultra-dense cellular networks. Black lines show the border of cell coverage. The gray area is plotted to refer to cell-edge area where JT can be supported.

conclusions in Section VI. The notations used in this paper are described in Table I.

II. System Model

In this section, we introduce a cache-enabled CoMP-integrated UDCN and give an overview of the multi-dimensional cooperative caching in the network.

A. Network Description and Modeling

We consider a UDCN consisting of a number of BSs and MTs, as illustrated in Fig. 1. The network is divided into cell-core and cell-edge areas according to the *long-term averaged* signal to interference plus noise ratio (SINR) of MTs. The throughput in the cell-edge area is generally lower than that in the cell-core area, due to the signal attenuation and interference caused by concurrent transmissions of other BSs [24]. We assume that all the BSs in the network support both ST and JT. The MTs located in the cell-core area are served with ST. The MTs in the cell-edge area can obtain not only ST service, but also JT service to improve throughput.

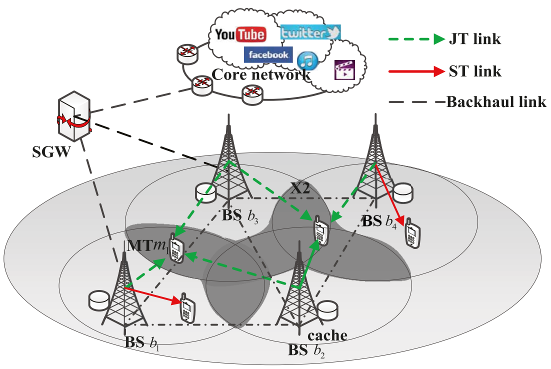


Fig. 2. Illustration of cooperative caching in CoMP-integrated system.

The caching and transmission network is shown in Fig. 2, where the content providers (CPs) (e.g. YouTube, Facebook, and so on) provide the library of multimedia files throughout the whole network. There are B BSs, denoted by $\mathcal{B} = \{b_1, \ldots b_i, \ldots, b_B\}$, under the centralized control of a service gateway (SGW). Each BS is able to cache some files with certain storage capacity to satisfy the requests from the MTs, and the cache capacity of BS b_i is denoted by C_{b_i}. The BSs in the network are connected with each other through optical fibers. To specify the topology of the BSs, we use the binary variable δ_{ir} to indicate whether BSs b_i and b_r are neighbors. If yes, $\delta_{ir} = 1$; otherwise $\delta_{ir} = 0$. Thus, the neighboring BSs of b_i, denoted by $\mathcal{N}_{b_i}$, can be given by

$$\mathcal{N}_{b_i} = \{b_r \in \mathcal{B} : \delta_{ir} = 1, b_i \neq b_r\}. \tag{1}$$

We note that each pair of neighboring BSs are bidirectionally connected with X2 interfaces, i,e., $\delta_{ir} = \delta_{ri}$, and the value of δ_{ir} is known once the network topology is given.

Moreover, there are M_i randomly distributed MTs, denoted by $\mathcal{M}_{b_i} = \{m_1, \ldots, m_j, \ldots, m_{M_i}\}$, in the cell of BS b_i, and $\mathcal{M}_{b_i}$ changes as the MTs enter or leave the cell. Denote g_{b_i,m_j} as the received SINR at MT m_j. We consider a user partitioning strategy in which BS b_i can judge whether a MT is at cell core or edge according to the long-term averaged SINR $\overline{g}_{b_i,m_j}$, which is denoted by

$$\overline{g}_{b_i,m_j} = \mathbb{E}_t(g_{b_i,m_j}) = \mathbb{E}_{t\in T}(\frac{\rho_i v_{ij} h_{ij}^{-\alpha}}{\sigma^2 + I_h}), \tag{2}$$

where T is the time length for updating g_{b_i,m_j}, ρ_i is the transmit power of BS b_i, $h_{ij}^{-\alpha}$ is the path loss from BS b_i to MT m_j, v_{ij} is the channel power gain, σ^2 denotes the noise power, and I_j is the experienced interference power at MT m_j. Then, the set of cell-edge MTs associated with BS b_i can be determined as follows:

$$\mathcal{M}_{b_i}^{ed} = \{m_j \in \mathcal{M}_{b_i} : \overline{g}_{b_i,m_j} \leq \tau\}, \tag{3}$$

where τ is the SINR threshold for performing a qualified ST. Similarly, the set of cell-core MTs that hold $\overline{g}_{b_i,m_j} >$

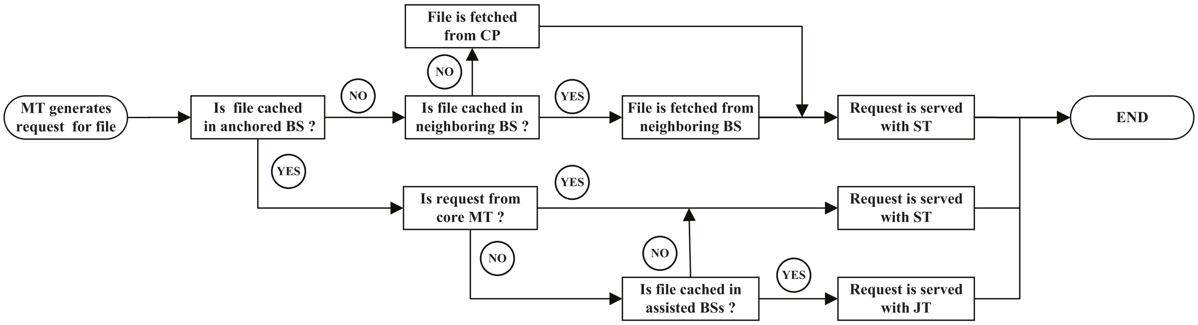

MT generates request for file
Is file cached in anchored BS ?
NO
Is file cached in neighboring BS ?
NO
File is fetched from CP
YES
File is fetched from neighboring BS
Request is served with ST
END
YES
Is request from core MT ?
YES
Request is served with ST
NO
NO
Is file cached in assisted BSs ?
YES
Request is served with JT

Fig. 3. Process of MTs that obtains desired files in CoMP-integrated cellular networks.

τ is denoted by $\mathcal{M}_{b_i}^{co}$. BS b_i updates sets $\mathcal{M}_{b_i}^{ed}$ and $\mathcal{M}_{b_i}^{co}$ periodically according to the reported $\overline{g}_{b_i,m_j}$. We consider a BS association strategy that each MT in the network is only associated with the BS which has the biggest $\overline{g}_{b_i,m_j}$, and the BS is called as *anchored-BS*. For the cell-edge MTs, the other BSs that can be reached by them are called as *assisted-BSs*. The assisted-BSs coordinate with the anchored-BS to provide JT (e.g., in Fig. 2, for MT m_1, BS b_1 is the anchored-BS and BSs b_2, b_3 are the assisted-BSs.).

B. Request and File Characteristics

We consider a finite file library, denoted by $\mathcal{F} = \{f_1, \ldots, f_k, \ldots, f_F\}$, consisting of F different files, and the size of file f_k is S_{f_k}. Each file has a content popularity, which is denoted by $P_{f_k}, \forall f_k \in \mathcal{F}$. As considered in previous works [25] and [26], the distribution of content popularity follows the generalized Zipf function and is obtained as

$$P_{f_k} = \left(rank(f_k)^{\alpha} \sum_{n=1}^{F} n^{-\alpha} \right)^{-1}, \quad \forall f_k \in \mathcal{F}, \tag{4}$$

where $0 \leq \alpha \leq 1$ is the skewness parameter characterizing the Zipf distribution. $\alpha = 0$ makes the distribution uniform and all the files have the same popularity, whereas $\alpha \geq 0$ makes P_{f_k} follow the classic Zipf's law implying that the files in $\mathcal{F}$ have uneven popularity. In this paper, we consider a popularity model, in which the files at different BSs have different local popularity distributions. Denote the local popularity of file f_k at BS b_i by P_{b_i,f_k}. Local popularity P_{b_i,f_k} indicates the user preference for file f_k at BS b_i. Clearly, we have $1 = \sum_{f_k \in \mathcal{F}} P_{f_k} = \sum_{b_i \in \mathcal{B}} \sum_{f_k \in \mathcal{F}} P_{b_i,f_k}$.

In addition, in this paper, the file request process is modeled as a Markov Modulated Rate Process (MMRP) [27]. That is to say, for MT m_j associated with BS b_i, the number of requests from it follows the Poisson process with mean rate λ_{b_i,m_j}. We assume that the probability of file f_k being requested at BS b_i is identical for each MT in the cell of BS b_i. Then, the average request arrival rate of file f_k from MT m_j, denoted by $\lambda_{b_i,m_j}^{f_k}$, can be obtained based on the normalized local popularity $\overline{P}_{b_i,f_k}$. Accordingly, we have

$$\begin{aligned} \overline{P}_{b_i,f_k} &= \frac{P_{b_i,f_k}}{\sum\limits_{f_k \in \mathcal{F}} P_{b_i,f_k}}, \quad \forall b_i \in \mathcal{B}, \ \forall f_k \in \mathcal{F}, \\ \lambda_{b_i,m_j}^{f_k} &= \lambda_{b_i,m_j} \overline{P}_{b_i,f_k}, \quad \forall b_i \in \mathcal{B}, \ \forall m_j \in \mathcal{M}_{b_i}, \ \forall f_k \in \mathcal{F}. \end{aligned} \tag{5}$$

C. Multidimensional Cooperative Caching

In the conventional caching schemes, MTs obtain the desired files from the cache-enabled BSs through ST. Instead, in the CoMP-integrated system, we propose a content delivery scheme that each MT can obtain a file either from its anchored-BS by ST or from both anchored BS and assisted-BSs by JT, based on the caching state of the file. Fig. 3 shows how the MT obtains its desired file using different transmission methods based on the files' placement in the CoMP-integrated UDCN. To make the caching efficient and adaptive to the hybrid transmission, we consider an integrated caching scheme with the following two types of cooperation:

Storage-Dimension Cooperation: The BSs in the network make caching decisions according to the average request rate of each file. The cached files are updated according to the changes of MTs' requests. To achieve efficient cooperation between BSs, the caching information and the cached files are shared between adjacent BSs through the high-capacity X2 interfaces. Accordingly, for a certain MT (in cell-core or cell-edge area) served with ST, the request from it can be satisfied by its anchored-BS directly (if the anchored-BS has cached the desired file) or through a neighboring BS indirectly (which is inaccessible for the MT but has the file cached). Otherwise, the desired file will be fetched from the CP. The details of the request routing strategy are described in following Section IV. In this context, to ensure high cache hit rate, two aspects are considered when deploying caching: (1) what content needs to be cached by the anchored-BS; (2) what content needs to be cached as a supplement in the neighboring BSs.

Transmission-Dimension Cooperation: In the cell-edge area, when a MT requests for a file, the BSs decide the transmission method (i.e., ST or JT) depending on the caching state of the desired file in the potential serving BSs. If the file is cached by multiple BSs (i.e., the anchored-BS and the assisted-BSs), the MT will be served with JT. Otherwise, i.e., the file is only cached by the anchored-BS or an assisted-BS, the request will be satisfied by ST.[1] Since JT can provide higher throughput for the cell-edge MTs compared with ST, the BSs are encouraged to cache the same files to facilitate more JT opportunities.

[1] Note that another opportunity for using JT is to wait the BS (anchored or assisted) which does not has the desired file to download it from a cooperative BS or the CP. However, compared with the case that the file has been ready for CoMP-JT, it needs extra delay and backhaul load. So we do not consider it as an option.

III. Delay Analysis and Problem Formulation

In this section, we analyze the content delivery delay under the proposed caching strategy and formulate the optimization problem. Then, the problem is transformed with approximation and substitution to reduce the complexity.

A. Delay Analysis

We incorporate the two types of cooperation into a multidimensional cooperative caching scheme, and formulate the optimization problem for the purpose of minimizing the content delivery delay. Note that a file is assumed to be either entirely cached or not cached in the network. The calculated delay for MTs to obtain the desired contents depends on the caching states of BSs, the selected transmitting methods, and the content sharing between BSs. We use three binary variables x_{b_i,f_k}, z_{b_i,f_k}, and $y^{f_k}_{b_i,b_r}$ to denote the caching and transmission decisions in the network, which are expressed as follows:

$$x_{b_i,f_k} = \begin{cases} 1 & \text{if file } f_k \text{ is cached by BS } b_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{b_i,f_k} = \begin{cases} 1 & \text{if BS } b_i \text{ transmits file } f_k \text{ with JT}, \\ 0 & \text{otherwise.} \end{cases}$$

$$y^{f_k}_{b_i,b_r} = \begin{cases} 1 & \text{if BS } b_r \text{ delivers file } f_k \text{ to BS } b_i, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we have the caching decision profile $\boldsymbol{x} = \{x_{b_i,f_k}\}_{b_i\in\mathcal{B},f_k\in\mathcal{F}}$, the JT decision profile $\boldsymbol{z} = \{z_{b_i,f_k}\}_{b_i\in\mathcal{B},f_k\in\mathcal{F}}$, and the routing decision profile $\boldsymbol{y} = \{y^{f_k}_{b_i,b_r}\}_{b_i\in\mathcal{B},b_r\in\mathcal{N}_{b_i},f_k\in\mathcal{F}}$.

In the CoMP-integrated UDCN, the content delivery delay is composed of the radio access delay and the backhaul transmission delay. To quantify the content delivery in the radio access network, we denote $r^{\text{ST}}_{b_i,m_j}$ as the transmit rate from BS b_i to MT m_j with ST. Then, the transmit delay $d^{\text{ST}}_{b_i,m_j,f_k}$ for MT m_j to obtain file f_k can be expressed as

$$d^{\text{ST}}_{b_i,m_j,f_k} = \frac{S_{f_k}}{r^{\text{ST}}_{b_i,m_j}} = \frac{S_{f_k}}{W\log_2(1+g_{b_i,m_j})}, \tag{6}$$

where W is the spectrum bandwidth, and $r^{\text{ST}}_{b_i,m_j}$ is obtained according to Shannon bound. Correspondingly, we denote $r^{\text{JT}}_{\mathcal{A},m_j}$ as the transmit rate of a JT from the BSs in $\mathcal{A}_{m_j}$ to MT m_j, where $\mathcal{A}_{m_j}$ is the set of BSs that can perform JT for MT m_j. Then, the transmit delay $d^{\text{JT}}_{\mathcal{A},m_j,f_k}$ for MT m_j to obtain file f_k can be expressed as

$$d^{\text{JT}}_{\mathcal{A},m_j,f_k} = \frac{S_{f_k}}{r^{\text{JT}}_{\mathcal{A},m_j}} = \frac{S_{f_k}}{W\log_2(1+\sum\limits_{b_i\in\mathcal{A}_{m_j}} g_{b_i,m_j})}. \tag{7}$$

For content delivery in the backhaul network, we denote the average transmit rate from neighboring BS b_r to current BS b_i by $r^{\text{NE}}_{b_i,b_r}$. Then, the average delay for BS b_i to fetch file f_k from BS b_r can obtained as $d^{\text{NE}}_{b_i,b_r,f_k} = S_{f_k}/r^{\text{NE}}_{b_i,b_r}$. Similarly, the average delay for BS b_i to fetch file f_k from the CP is obtained as $d^{\text{CP}}_{b_i,f_k} = S_{f_k}/r^{\text{CP}}_{b_i}$, where $r^{\text{CP}}_{b_i}$ is the average transmit rate in the core network.

Accordingly, based on the caching and transmitting decisions, we consider the following three cases to calculate the content delivery delay:

Case 1: For MT m_j associated with BS b_i, if its desired file f_k has been cached at anchored-BS b_i or the assisted BSs, the MT can obtain file f_k from the local BSs with ST or JT directly, depending on the caching state of the file. The total delay for all the MTs to access file f_k with ST and JT intra the cell of BS b_i is calculated as $\mathcal{D}^{intra}_{b_i,f_k}(x_{b_i,f_k}, z_{b_i,f_k})$, which is given as

$$\begin{aligned} &\mathcal{D}^{intra}_{b_i,f_k}(x_{b_i,f_k}, z_{b_i,f_k}) \\ &\quad = x_{b_i,f_k} \sum_{m_j\in\mathcal{M}^{co}_{b_i}} \lambda^{f_k}_{b_i,m_j} d^{\text{ST}}_{b_i,m_j,f_k} \\ &\qquad + \sum_{m_j\in\mathcal{M}^{ed}_{b_i}} \lambda^{f_k}_{b_i,m_j} \Big(z_{b_i,f_k} d^{\text{JT}}_{\mathcal{A},m_j,f_k} + q_{b_i,f_k} d^{\text{ST}}_{b_i,m_j,f_k} \Big), \end{aligned} \tag{8}$$

where $q_{b_i,f_k} = x_{b_i,f_k} - z_{b_i,f_k}$. Note that BS b_i can perform JT for file f_k (i.e., $z_{b_i,f_k} = 1$) on the premises that 1) file f_k has been cached at BS b_i (i.e., $x_{b_i,f_k} = 1$, and $z_{b_i,f_k} \leq x_{b_i,f_k}$), and 2) each BS in $\mathcal{A}_{m_j}$ has a same JT decison (i.e., $z_{b_i,f_k} = z_{b_n,f_k} = 1, \forall b_n \in \mathcal{A}_{m_j}$). Assuming that the storage capacity at each BS is large enough to cache all the files in $\mathcal{F}$, then all the cell-core requests can be satisfied by ST and all the cell-edge requests can be satisfied by JT. However, in fact, the cache capacity of a BS is limited and the sum size of all the cached files cannot exceed the total storage capability of the BS, i.e., $\sum_{f_k\in\mathcal{F}} S_{f_k} x_{b_i,f_k} \leq C_{b_i}$ must hold.

Case 2: If file f_k is not cached at BS b_i but has been cached by a neighboring BS, BS b_i first fetches file f_k from the neighboring BS and then transmits it to a MT with ST. The total delay for all the MTs to obtain file f_k by inter-BS cooperation is calculated as $\mathcal{D}^{inter}_{b_i,f_k}(y^{f_k}_{b_i,b_r})$, which is given by:

$$\begin{aligned} &\mathcal{D}^{inter}_{b_i,f_k}(y^{f_k}_{b_i,b_r}) \\ &\quad = \sum_{b_r\in\mathcal{N}_{b_i}} \sum_{m_j\in\mathcal{M}_{b_i}} y^{f_k}_{b_i,b_r} \lambda^{f_k}_{b_i,m_j} \Big(d^{\text{NE}}_{b_i,b_r,f_k} + d^{\text{ST}}_{b_i,m_j,f_k} \Big). \end{aligned} \tag{9}$$

The neighboring BS b_r decides to transmit file f_k to BS b_i on the premise that file f_k has been cached by itself (i.e., $x_{b_r,f_k} = 1$, and $y^{f_k}_{b_i,b_r} \leq x_{b_r,f_k}$). Meanwhile, it is required that the content transmission among the neighboring BSs of BS b_i is not redundant, and any file will not be routed to BS b_i if the file is locally available. In other words, $\sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r} + x_{b_i,f_k} \leq 1$ must hold.

Case 3: If file f_k cannot be fetched from the caches in the RAN, the total delay for all the MTs to obtain file f_k from the CP in the core network is calculated as $\mathcal{D}^{outer}_{b_i,f_k}(x_{b_i,f_k}, y^{f_k}_{b_i,b_r})$. We have

$$\begin{aligned} &\mathcal{D}^{outer}_{b_i,f_k}(x_{b_i,f_k}, y^{f_k}_{b_i,b_r}) \\ &\quad = \Bigg(1 - x_{b_i,f_k} - \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r} \Bigg) \sum_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} o_{b_i,m_j,f_k}, \end{aligned} \tag{10}$$

where $o_{b_i,m_j,f_k} = d^{\text{CP}}_{b_i,f_k} + d^{\text{ST}}_{b_i,m_j,f_k}$.

B. Optimization Problem

Based on the discussions above, we formulate the cooperative caching problem. Since the goal is to minimize the content delivery delay for both cell-core and cell-edge MTs simultaneously, we jointly search the optimal values of $\{x_{b_i,f_k}, y_{b_i,b_r,f_k}, z_{b_i,f_k}\}$ under the cache capacity limits.

Let

$$\Psi(x_{b_i,f_k}, y^{f_k}_{b_i,b_r}, z_{b_i,f_k}) = \mathcal{D}^{intra}_{b_i,f_k}(x_{b_i,f_k}, z_{b_i,f_k}) + \mathcal{D}^{inter}_{b_i,f_k}(y^{f_k}_{b_i,b_r}) + \mathcal{D}^{outer}_{b_i,f_k}(x_{b_i,f_k}, y^{f_k}_{b_i,b_r}), \quad (11)$$

then, the overall problem is formulated as follows:

Problem (I)

$$\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}{\text{Min}} \ & \sum_{b_i\in\mathcal{B}}\sum_{f_k\in\mathcal{F}} \Psi(x_{b_i,f_k}, y^{f_k}_{b_i,b_r}, z_{b_i,f_k}) \\
\text{s.t. } c1:\ & \sum_{f_k\in\mathcal{F}} S_{f_k} x_{b_i,f_k} \le C_{b_i}, \quad \forall b_i, \\
c2:\ & x_{b_i,f_k} - z_{b_i,f_k} \ge 0, \quad \forall b_i,\ \forall f_k, \\
c3:\ & z_{b_i,f_k} = z_{b_r,f_k}, \quad \forall b_i, \forall b_r \in \mathcal{A}_{m_j},\ \forall f_k, \\
c4:\ & y^{f_k}_{b_i,b_r} \le x_{b_r,f_k}, \quad \forall b_i,\ \forall b_r \in \mathcal{N}_{b_i}, \forall f_k, \\
c5:\ & x_{b_i,f_k} + \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r} \le 1, \quad \forall b_i,\ \forall f_k, \\
c6:\ & x_{b_i,f_k}, z_{b_i,f_k} \in \{0,1\}, \quad \forall b_i, \forall f_k, \\
c7:\ & y^{f_k}_{b_i,b_r} \in \{0,1\}, \quad \forall b_i, \forall b_r \in \mathcal{N}_{b_i},\ \forall f_k, \quad (12)
\end{aligned}$$

where constraint $c1$ guarantees that the sum size of the cached files does not exceed the total storage space, constraint $c2$ means that the JT decision of BS b_i is restricted by the caching state of file f_k, constraint $c3$ guarantees that all the BSs in set $\mathcal{A}_{m_j}$ hold the same JT decision regarding the delivery of file f_k to MT m_j, constraint $c4$ denotes that a neighboring BS will not be able to transmit the desired file to the requesting BS if the file is not locally cached, constraint $c5$ is proposed to ensure that the content sharing between BSs is performed only when file f_k is not cached at BS b_i. Meanwhile, only one neighboring BS is allowed to transmit file f_k to BS b_i at a certain time to avoid transmission redundancy.

C. Problem Transformation

Problem (I) is a BILP problem, which is NP-hard due to the following reasons:

- The feasible set of Problem (I) is not convex since the elements in sets $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are binary variables. The complexity of the objective function exponentially increases with the space size of $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, together with binary characteristic variables and the inequalities of constraints $c1$, $c2$, $c4$, and $c5$. The BILP problem can be reduced to either the Travel Saleman problem (TSP) or the Partition problem[29][30], which is proved to be NP-hard.
- The number of BSs and the number of files in practice make the solution space of sets $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ become very large. If the numbers of BSs and files are respectively B and F, the number of variables in Problem (I) will reach $2FB+B^2F$. Correspondingly, an algorithm to search the optimal solution will hold the complexity of $\mathcal{O}((2FB+B^2F)^l)$ ($l=1$ implies a linear algorithm while $l>1$ denotes a polynomial time algorithm).

To solve this problem and obtain feasible lower-complexity solutions, a transformation and simplification of the original problem are necessary.

Through algebraic substitution, the optimization objective of Problem (I) can be rewritten as

$$\begin{aligned}
&\Psi(x_{b_i,f_k}, y^{f_k}_{b_i,b_r}, z_{b_i,f_k}) \\
&= x_{b_i,f_k} \sum_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} d^{\text{ST}}_{b_i,m_j,f_k} \\
&\quad + z_{b_i,f_k} \sum_{m_j\in\mathcal{M}^{ed}_{b_i}} \lambda^{f_k}_{b_i,m_j} \left(d^{\text{JT}}_{b_i,m_j,f_k} - d^{\text{ST}}_{b_i,m_j,f_k}\right) \\
&\quad + \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r} \sum_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} \left(d^{\text{NE}}_{b_i,b_r,f_k} + d^{\text{ST}}_{b_i,m_j,f_k}\right) \\
&\quad + \left(1 - x_{b_i,f_k} - \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r}\right) \sum_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} o_{b_i,m_j,f_k} \\
&= z_{b_i,f_k}\zeta_{b_i,f_k} + \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r}\phi_{b_i,f_k} \\
&\quad + \left(1 - x_{b_i,f_k} - \sum_{b_r\in\mathcal{N}_{b_i}} y^{f_k}_{b_i,b_r}\right)\xi_{b_i,f_k} + w_{b_i,f_k}, \quad (13)
\end{aligned}$$

where

$$\begin{cases}
\zeta_{b_i,f_k} = \sum\limits_{m_j\in\mathcal{M}^{ed}_{b_i}} \lambda^{f_k}_{b_i,m_j} \left(d^{\text{JT}}_{b_i,m_j,f_k} - d^{\text{ST}}_{b_i,m_j,f_k}\right) \\
\phi_{b_i,f_k} = \sum\limits_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} d^{\text{NE}}_{b_i,b_r,f_k} \\
\xi_{b_i,f_k} = \sum\limits_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} d^{\text{CP}}_{b_i,f_k} \\
w_{b_i,f_k} = \sum\limits_{m_j\in\mathcal{M}_{b_i}} \lambda^{f_k}_{b_i,m_j} d^{\text{ST}}_{b_i,m_j,f_k}.
\end{cases}$$

Then, based on constraints $c4$ and $c5$ in Problem (I), the optimization objective (13) can be equivalent to

$$\begin{aligned}
&\Psi(x_{b_i,f_k}, z_{b_i,f_k}) \\
&\quad = z_{b_i,f_k}\zeta_{b_i,f_k} + \left(\bigcup_{b_r\in\mathcal{N}_{b_i}} x_{b_r,f_k} - x_{b_i,f_k}\right)\phi_{b_i,f_k} \\
&\qquad + \left(1 - \bigcup_{b_r\in\mathcal{N}_{b_i}\cup b_i} x_{b_r,f_k}\right)\xi_{b_i,f_k} + w_{b_i,f_k}, \quad (14)
\end{aligned}$$

where the logical operation $\bigcup$ indicates 'OR' operation.

Proof: See Appendix A. ■

Then, Problem I is converted into the optimization problem for content placement x_{b_i,f_k} and transmit method z_{b_i,f_k} as follows

Problem (II)

$$\begin{aligned}
&\underset{\boldsymbol{x},\boldsymbol{z}}{\text{Min}} \sum_{b_i\in\mathcal{B}}\sum_{f_k\in\mathcal{F}} \Psi(x_{b_i,f_k}, z_{b_i,f_k}) \\
&\text{s.t. } c1, c2, c3, c6. \quad (15)
\end{aligned}$$

The content sharing decision $\boldsymbol{y}$ can be obtained under the achieved $\boldsymbol{x}$, which is formulated in Algorithm 2.

Lemma 1: For any solutions $(x_{b_i,f_k})^*$ and $(z_{b_i,f_k})^*$ under the constraints of $c2$ and $c3$ in Problem (II), if they are local or global optimal, they will satisfy:

$$\left.\begin{matrix}(z_{b_i,f_k})^* \\ s.t.\ C2, C3\end{matrix}\right\} = \bigcap_{b_r \in \mathcal{A}_{m_j}} (x_{b_r,f_k})^*, \quad \forall b_i \in \mathcal{B}, f_k \in \mathcal{F}, \quad (16)$$

where the logical operation $\bigcap$ indicates 'AND' operation.

Proof: See Appendix B. ■

Based on Lemma 1, plugging (16) into (14), the objective of Problem (II) can be rewritten as

$$\begin{aligned}&\Psi(x_{b_i,f_k}) \\ &= \bigcap_{b_r \in \mathcal{N}_{b_i} \cup \{b_i\}} (x_{b_i,f_k})\,\zeta_{ik} + \left(\bigcup_{b_r \in \mathcal{N}_{b_i}} x_{b_r,f_k} - x_{b_i,f_k}\right)\phi_{b_i,f_k} \\ &\quad + \left(1 - \bigcup_{b_r \in \mathcal{N}_{b_i} \cup \{b_i\}} x_{b_r,f_k}\right)\xi_{b_i,f_k} + w_{b_i,f_k}. \qquad (17)\end{aligned}$$

Based on the above analysis, Problem (II) is transformed into

Problem (III)

$$\underset{\boldsymbol{x}}{\text{Min}} \;\; \Psi(x_{b_i,f_k}) \qquad (18a)$$

$$\text{s.t.} \;\; \sum_{f_k \in \mathcal{M}} S_{f_k} x_{b_i,f_k} \le C_{b_i}, \quad \forall b_i \in \mathcal{B}, \qquad (18b)$$

$$x_{b_i,f_k} \in \{0,1\}, \quad \forall b_i \in \mathcal{B},\; f_k \in \mathcal{F}. \qquad (18c)$$

Note that Problem (II) is still NP-hard, and it is difficult to obtain a closed form solution. In the next section, we will develop a heuristic algorithm based on Genetic Algorithm (GA) to solve the problem.

IV. Optimization Algorithm Design

To obtain the optimal solution, a straightforward method is to conduct exhaustive search by testing all the possible file placement x_{b_i,f_k}, $\forall b_i \in \mathcal{B}$, $\forall f_k \in \mathcal{F}$. However, it is infeasible because a big number of files make the computation of an optimal solution impossible in practice. We are motivated to develop an algorithm that is effective and can be easily implemented by network operator. The heuristic search algorithm GA is considered to be suitable for massively parallel optimization [28], [29]. GA provides a robust, near-optimal solution for many real-world NP-hard problems such as BS placement in heterogeneous networks [30], video chunks placement in CDN [31], channel subcarrier assignment [29], [32]. On our side, we choose GA because its parallelization is just suitable to our problem. In the following, we present how we solve the problem with GA, and then introduce a file updating and scheduling strategy.

A. Background on Genetic Algorithm

GA is a stochastic optimization method inspired by the evolution theory. It mimics the process of natural selection [28]. GA starts with a randomly generated population consisting of multiple different individuals from the entire range of possible individuals. The individuals are transformed into the genetic form (usually digital numbers) by an encoding method. Then, two steps are repeated to address the population during the evolution.

1) The fitness of each individual is evaluated by a fitness function so that the elite individuals with high fitness can be selected as "parents" by a selection function.
2) Genetic operators, *crossover* and *mutation*, operate the "parent" individuals to produce m offsprings to evolve the current generation into a new one.

The evolution process will repeat until a termination condition is reached. In this way, the initial population is updated, making the individual matrices denser around the optimal solution.

B. Individual Construction

The caching decision is in binary, therefore the file placement result itself could be chromosome and there is no need to encode the individuals. For an B-BS and F-file caching network, the chromosome of an individual can be a $B \times F$ binary matrix, which is given by

$$X_{B\times F} = \begin{bmatrix} x_{1,1}, & \cdots, & x_{1,F} \\ \vdots & \vdots & \vdots \\ x_{B,1}, & \cdots, & x_{B,F} \end{bmatrix}. \qquad (19)$$

We set that the initial population has N_{pop} individual matrices. Initialization of the individuals is described by Steps 2 to 9 in Algorithm 1. Specifically, for each matrix $X^{B\times F}$ of the first $N_{pop} - 2$ individual matrices, we select the first U entries of each row as the valid chromosomes based on the fact that there is no benefit to cache the files with lower popularity, and U satisfies

$$U = \sum_{i=1}^{B} N(C_{b_i}) \le F, \qquad (20)$$

where $N(C_{b_i})$ denotes the number of files that can be cached given the storage size C_{b_i}. Then, randomly select $N(C_{b_i})$ of the U valid entries and set them to be one. The remaining entries are set to be zero. Then, the last two individual matrices processed by the MPC and LCD schemes are added into the initial population to improve convergence rate.

C. Fitness Evaluation and Selection

In GA, the fitness function is the key to driving the search towards a promising trend and finally finding the optimal solution. The fitness value of a certain individual is computed through the fitness function with the values of $\{x_{b_i,f_k}\}$. In Problem (III), the objective function can be expressed as a function of file placement array $X_{b_i}^{1\times F} = [x_{b_i,f_1}, \ldots, x_{b_i,f_k}]$. Therefore, we conduct the fitness function as follows

$$\mathcal{G}(X) = \sum\nolimits_{b_i \in \mathcal{B}} \widetilde{\Psi}(X_{b_i}^{1\times F}). \qquad (21)$$

Based on (21), the fitness values of N_{pop} individuals are evaluated.

For the selection process, this paper adopts the famous *tournament selection* method to select the elite individuals. Specifically, in each selection the method randomly picks

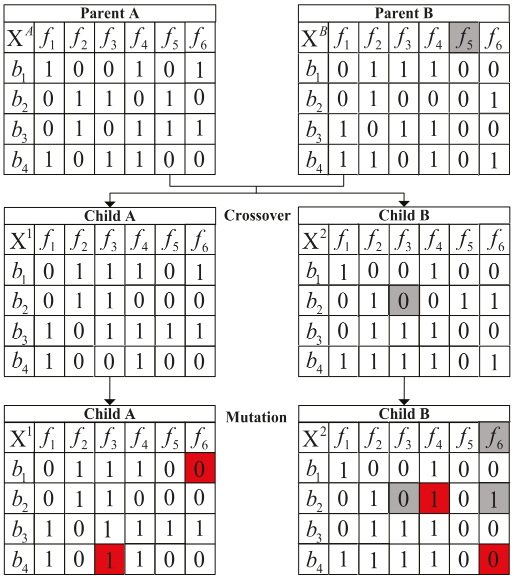
Algorithm 1 GA-Based Joint Optimization for Content Caching

```
1: Input: $N_{pop}$, $N_{ele}$, $p_{cro}$, $p_{mut}$, $\varepsilon$.
2: Initialize: $X_n = 0^{B \times F}$ for $\forall n \in [1, N_{pop}]$, $Q = \emptyset$, $E = \emptyset$, $U = \sum_{i=1}^{B} N(C_{b_i})$.
3: For each row of matrix $X_n, n \in [1, N_{pop} - 2]$, randomly set $N(C_{b_i})$ of the first $U$ bits to be 1 and $Q = Q \cup \{X_n\}$.
4: Implement matrix $X_{N_{pop}-1}$ with MPC strategy and matrix $X_{N_{pop}}$ with LCD strategy.
5: $Q = Q \cup \{X_{N_{pop}-1}, X_{N_{pop}}\}$.
6: repeat
7:    Compute the fitness value of $X_n, \forall X_n \in Q_{current}$.
8:    Retain the first $N_{ele}$ matrices in $Q_{current}$ in descending order of $\mathcal{G}_{current}(X_n)$, and copy them into $E$.
9:    $Q_{new} = Q_{new} \cup E$.
10:   while $sum\,(Q_{new}) < N_{pop}$ do
11:       Get matrices $X_a$ and $X_b$ from $E$.
12:       if $random[0, 1] < p_{cro}$ then
13:           Cross $X_a$ and $X_b$ using multifragment parallel crossover strategy and get children $X'_a$ and $X'_b$.
14:       end if
15:       if $random[0, 1] < p_{mut}$ then
16:           Mutate $X'_a$ and $X'_b$ using the mutation strategy.
17:       end if
18:       while $\sum_{f_k \in \mathcal{F}} x_{b_i,f_k} S_{f_k} > N(C_{b_i})$, $x_{b_i,f_k} \in X'_a$ or $X'_b$ do
19:           Set $x_{b_i,f_k}$ to 0 in ascending order of $p_{b_i,f_k}$.
20:       end while
21:       while $\sum_{k \in \mathcal{F}} x_{b_i,f_k} < N(C_{b_i})$, $x_{b_i,f_k} \in X'_a$ or $X'_b$ do
22:           Set $x_{b_i,f_k}$ to 1 in descending order of $p_{b_i,f_k}$.
23:       end while
24:       $Q_{new} = Q_{new} \cup \{X'_a, X'_b\}$.
25:   end while
26:   $\gamma = |(\mathcal{G}_{new}(X) - \mathcal{G}_{current}(X))/\mathcal{G}_{current}(X)|$.
27:   $Q_{current} = Q_{new}$, $Q_{new} = \emptyset$, $E = \emptyset$.
28: until $\gamma < \varepsilon$
29: Output: $X_n^*$ which has the biggest $\mathcal{G}(X_n^*)$ in $Q_{new}$.
```

R individuals into a tournament group, and selects the one with the best fitness value into the next generation as the parent individual. This process repeats until the parent group is enough. In algorithm 1, to reduce the complexity, we select the first N_{ele} individuals in descending order of the fitness values and copy them directly into the next generation as parent individuals. The evaluation and selection processes are described by Steps 10 to 20 in Algorithm 1.

D. Crossover and Mutation

The crossover is used to mix chromosomes between individuals to produce more fitter offsprings. In our algorithm, to increase the crossover efficiency, we design a *multi-fragment parallel crossover* method to produce individuals. The crossover is operated on the parents with probability p_{cro}. The parallel crossovers are performed on multiple pairs of two rows, each of which is from two individual matrixes. Then multiple chromosome fragments from multiple rows are switched concurrently between two parent individuals to produce two offspring individuals. The mutation is used to add some randomness to the evolution process to avoid the solutions from local optima. During the mutation process, the bits at random locations of the offspring individuals are turned over with low probability p_{mut}. The crossover and mutation processes are described by steps 13 to 17 in Algorithm 1. Considering the storage constraint may be violated after the crossover and mutation operations, the individual matrices are repaired to ensure that the size of cached file equals to the storage space of each BS(see steps 18 to 23).



Parent A

X^A	f_1	f_2	f_3	f_4	f_5	f_6
b_1	1	0	0	1	0	1
b_2	0	1	1	0	1	0
b_3	0	1	0	1	1	1
b_4	1	0	1	1	0	0

Parent B

X^B	f_1	f_2	f_3	f_4	f_5	f_6
b_1	0	1	1	1	0	0
b_2	0	1	0	0	0	1
b_3	1	0	1	1	0	0
b_4	1	1	0	1	0	1

Child A

X^1	f_1	f_2	f_3	f_4	f_5	f_6
b_1	0	1	1	1	0	1
b_2	0	1	1	0	0	0
b_3	1	0	1	1	1	1
b_4	1	0	0	1	0	0

Child B

X^2	f_1	f_2	f_3	f_4	f_5	f_6
b_1	1	0	0	1	0	0
b_2	0	1	0	0	1	1
b_3	0	1	1	1	0	0
b_4	1	1	1	1	0	1

Child A

X^1	f_1	f_2	f_3	f_4	f_5	f_6
b_1	0	1	1	1	0	0
b_2	0	1	1	0	0	0
b_3	1	0	1	1	1	1
b_4	1	0	1	1	0	0

Child B

X^2	f_1	f_2	f_3	f_4	f_5	f_6
b_1	1	0	0	1	0	0
b_2	0	1	0	1	0	1
b_3	0	1	1	1	0	0
b_4	1	1	1	1	0	0

Fig. 4. Example of crossover and mutation with 4 BSs and 6 files.

One example of the crossover and mutation process is illustrated in Fig. 4, where the parents are constructed by a 4-BS and 6-file mapped matrix. The crossover points used to mark the chromosome fragments in each row are randomly generated as $e_{begin} = 1, 4, 1, 3$ and $e_{end} = 3, 5, 2, 5$. Then, the corresponding fragments are switched in the matrices of parents A and B. After crossover, children A and B are mutated by randomly turning over the binary values of the matrix.

The process of GA-based joint optimization for the content placement has been depicted in Algorithm 1. The complexity of the proposed GA-based algorithm is $\mathcal{O}(N_{pop}N_{gen})$, where N_{gen} is the number of generations evaluated (i.e., iteration times). The algorithm will finally converge to an optimal solution if the iteration keeps going on. However, it is time-consuming to obtain the optimal solution. In this case, the evolution can terminate in a finite number of iterations if the solution is near-optimal.

E. File Updating and Transmitting

Based on the above joint optimization of content placement, we can obtain the optimal caching decisions under the practical deployment of large-scale content distribution. To make the caching strategy work in an effective way, the cached files should be updated in time. Fig. 5 illustrates the updating process. The BS evaluates the local file request rate and sends the message to the computing center (CC) deployed in the SGW. The CC periodically solves the optimization problem

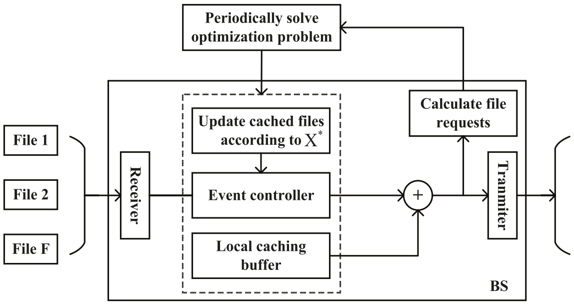


Fig. 5. Framework of file updating when caching is performed at a BS.

Algorithm 2 File Routing Strategy

```
1: **Input:** The optimal caching results $\{(x_{b_i,f_k})^*\}$.
2: **Wait until** receiving the request for file $f_k$ from MT $m_j$, then check the local caches.
3: **if** $x_{b_i,f_k} = 1$ **then**
4:     Satisfy the request at local BS with ST if $m_j \in \mathcal{M}_{b_i}^{co}$.
5:     Perform the following two options if $m_j \in \mathcal{M}_{b_i}^{ed}$:
6:     **a)**:route the request to the assisted-BSs $b_r, \forall b_r \in \mathcal{A}_{m_j}$
         and transmit file with JT, if $z_{b_i,f_k} = 1$, where $z_{b_i,f_k} = \bigcap_{b_r \in \mathcal{A}_{m_j}} x_{b_r,f_k}$.
7:     **b)**:Satisfy the request at local BS with ST, if $z_{b_i,f_k} = 0$.
8: **end if**
9: **if** $x_{b_i,f_k} = 0$ and $\bigcup_{b_r \in \mathcal{N}_{b_i}} x_{b_r,f_k} = 1$ **then**
10:    Set $y_{b_i,b_r^*}^{f_k} = 1$,
       where $b_r^* = \arg\min_{b_r \in \mathcal{N}_{b_i}} x_{b_r,f_k} \sum_{b_l \in \mathcal{N}_{b_r}} \sum_{m_j \in \mathcal{M}_{b_l}} \lambda_{b_l,m_j}^{f_k} S_{f_k}$.
11:    Route the request to BS $b_r^*$. Then $f_k$ is routed to BS $b_i$
       and the request is satisfied with ST.
12: **end if**
13: **if** $\bigcup_{b_r \in \mathcal{N}_{b_i} \cup \{b_i\}} x_{b_r,f_k} = 0$ **then**
14:    Route the request to the CP in core network, satisfy it
       with ST.
15: **end if**
```

based on the received messages, and distributes the optimal caching decisions to each BS. Then, during off-peak time, the file candidates to be cached are pre-fetched by BSs and the locally cached files are updated.

Given the optimal content placement of BSs, the routing decision $\boldsymbol{y}$ and JT decision $\boldsymbol{z}$ can be obtained under the achieved caching decision $\boldsymbol{x}$. The details of the routing and transmitting strategy are described in Algorithm 2. To obtain effective decisions $y_{b_i,b_r}^{f_k}, \forall b_r \in \mathcal{M}_{b_i}$, BS b_i selects a neighboring BS which has the lowest traffic load to delivery file f_k when file f_k is cached at multiple neighboring BSs (see Step 10). The JT decision for BS b_i to transmit f_k to MT m_j can be obtained as $z_{b_i,f_k} = \bigcap_{b_r \in \mathcal{A}_{m_j}} x_{b_r,f_k}$. Then, the transmit method in the RAN can be determined according to z_{b_i,f_k} (see steps 5 to 7).

V. Performance Evaluation

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Population size in GA N_{pop}	150
Number of elites N_{ele}	10
Crossover probability p_{cro}	0.95
Mutation probability p_{mut}	0.05
Terminate threshold ε	0.1%
Request arrival rate λ_{b_i,m_j}	10/sec
MT distribution η	[0.1, 0.8]
Number of MTs in each cell M	200
Size of file S_{f_k}	[8, 15] MB
Number of files F	2000
Number of BSs B	10
Wireless channel bandwidth W	5 MHZ
Transmit power of BSs	100 mWatts
Propagation path loss	$L(r)$ = 34 + 40log(r)
Background noise σ^2	-100dBm
SINR threshold τ	7 dB
Transmit rate r_{b_i,b_r}^{NE}, $r_{b_i}^{\text{CP}}$	[20,30] Mb/s, [5,8] Mb/s

In this section, the performance of the proposed MDCC scheme is evaluated through a series of simulations. We focus on analyzing the normalized content delivery latency (NorCDT) of our scheme over the existing caching schemes for the cell-core and cell-edge MTs, respectively. The NorCDT is given as

$$NorCDT = \frac{\mathcal{D}_{total}(Caching\_strategy)}{\mathcal{D}_{total}(No\_caching)} \tag{22}$$

where $\mathcal{D}_{total}(\cdot)$ means the sum delay for all the requests being satisfied under a certain caching strategy. To demonstrate how the multidimensional cooperation in caching impacts NorCDT in the CoMP-integrated UDCN, we apply the FemtoMPC and Advanced-FemtoMPC strategies as comparisons. The FemtoMPC and Advanced-FemtoMPC schemes are introduced as baseline models to demonstrate which improvement is contributed by JT.

1) *FemtoMPC*: This strategy is derived by modifying the FemtoCaching scheme [15]. The BSs in the neighborhood of a MT cache the most popular contents according to local popularity distribution until their spaces are full. Both storage-dimension and transmission-dimension cooperative cachings are not considered, and the MT can only be served by its anchored-BS with ST in the network.
2) *Advanced-FemtoMPC:* This strategy is an upgrade of the FemtoMPC strategy by adding a new function of exploiting JT opportunities into each BS, for the purpose of improving the cell-edge throughput.

In the simulations, we consider 10 BSs controlled by 1 SGW. The CP in the core network holds all the 2000 files. For simplicity, we assume that all the MTs are motionless during the simulation period unless stated otherwise. The MDCC, FemtoMPC, and Advanced-FemtoMPC strategies are implemented on the BSs respectively, and then we count the NorCDTs of cell-edge and cell-core MTs under different strategies. The parameter settings used in the simulations are summarized in Table II.

Fig. 6 shows the convergence behavior of the GA-based algorithm with the population size $N_{pop} = 150$, and elite size $N_{ele} = 10$. We observe that the fitness value drops sharply at the initial iterations, and then it converges after approximate 1000 generations.

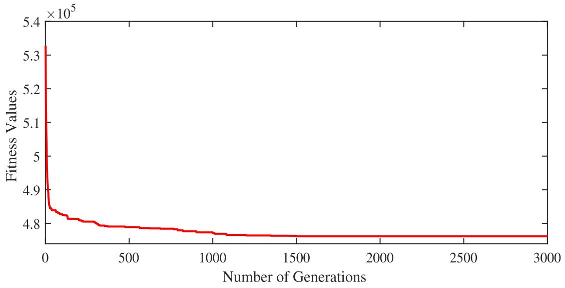
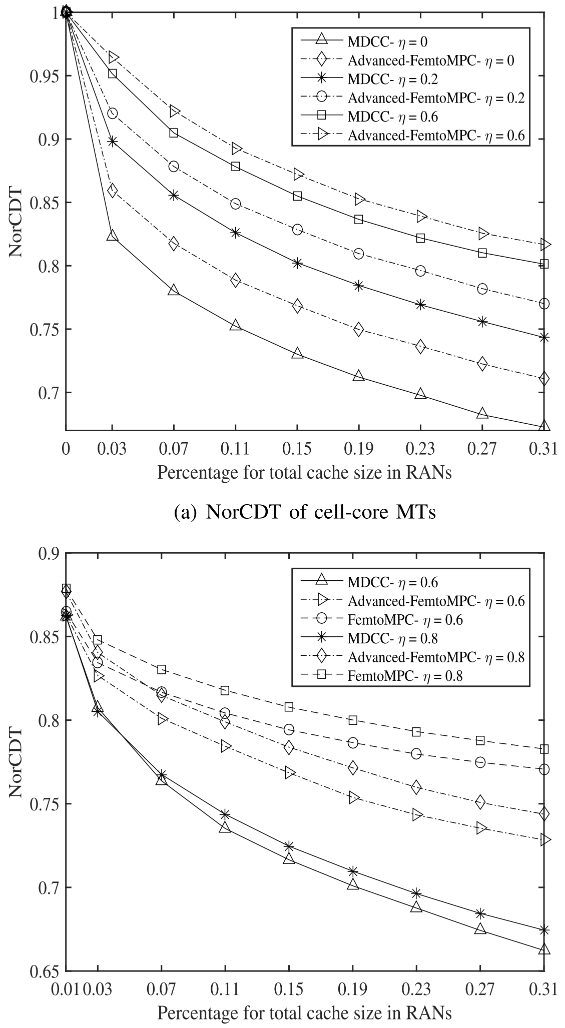


Fig. 6. Convergence behavior of the delay minimization algorithm.



(a) NorCDT of cell-core MTs

(b) NorCDT of cell-edge MTs

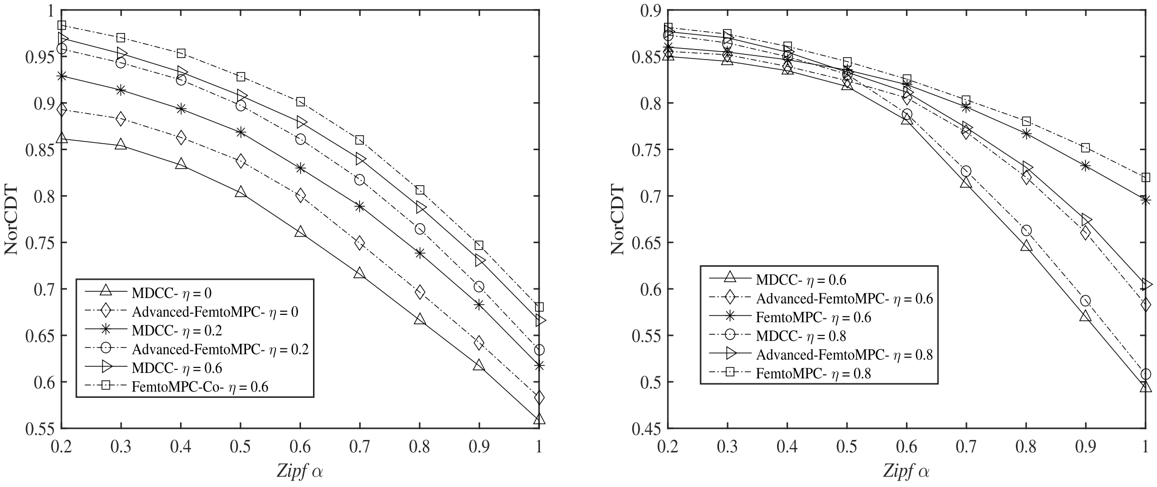
Fig. 7. NorCDTs of cell-core MTs (a) and cell-edge MTs (b) with respect to increasing cache size (percentage to the total content size) of the BSs in RANs, where Zipf $\alpha = 0.7$.

Fig. 7(a) depicts the NorCDT of the cell-core MTs changing with the total cache size of BSs for various user distribution ratio η, where η is the ratio of the number of cell-edge MTs to the number of cell-core MTs in a cell. A large value of η indicates there are more content requests from the cell-edge area. This figure demonstrates that the NorCDTs of the MDCC and Advanced-FemtoMPC schemes decrease with the increase of the total storage capacity of the network, and this can be explained by the fact that greater storage space make the BSs to cache more files, leading to higher cache hit rate. In addition, the MDCC scheme outperforms the Advanced-FemtoMPC scheme for different values of η, because the cooperation between neighboring BSs can squeeze more storage space than the isolated caching in the FemtoMPC scheme. For the extreme case that $\eta = 0$, i.e., all the MTs are in the cell-core area, the MDCC scheme allocates all the storage space to optimize the caching for ST. Hence, The MDCC scheme expresses the largest advantage compared with the cases that $\eta = 0.2, 0.6$. More importantly, since the MDCC scheme takes advantage of the joint optimization for ST and JT, it can sill hold better performance even when cell-edge MTs account for a large proportion in the network.

Fig. 7(b) shows the NorCDT of the cell-edge MTs versus the total cache size of BSs for various user distribution ratio η. Obviously, the Advanced-FemtoMPC scheme has lower NorCDT than the FemtoMPC scheme for the cell-edge MTs because the cell-edge MTs can be provided with JT service. Meanwhile, as expected, the MDCC scheme outperforms the Advanced-FemtoMPC scheme for various values of η. This is because the MDCC scheme appropriately allocates the caches used for ST and JT. The adjacent BSs are aspired to make enough copies of popular files to create JT opportunities, improving the throughput of cell-edge MTs.

Fig. 8(a) compares the NorCDT of cell-core MTs using the MDCC and Advanced-FemtoMPC schemes with Zipf parameter α varying from 0.2 to 0.9. We observe that the NorCDTs of both MDCC and Advanced-FemtoMPC schemes decrease sharply with the increase of Zipf parameter, since a more uneven popularity distribution can better reflect the advantage of the caching strategy. Fig. 8(b) shows the NorCDT of cell-edge MTs versus the Zipf parameter. We observe that the MDCC scheme always holds lower NorCDT than the Advanced-FemtoMPC scheme, especially when α is large. When α is small (ranging from 0.2 to 0.4), the MDCC scheme still holds advantages for cell-edge MTs compared with the FemtoMPC and Advanced-FemtoMPC schemes. As the value of α increases (ranging from 0.6 to 0.9), the jointly optimized caching for JT and ST works. The MDCC scheme pushes the BSs to make more copies of the more popular files and reduces the redundancy of the less popular ones between neighboring BSs, so that it can hold advantages for cell-edge MTs while guarantee the cache hit rate of cell-core MTs (see Fig. 8(a)).

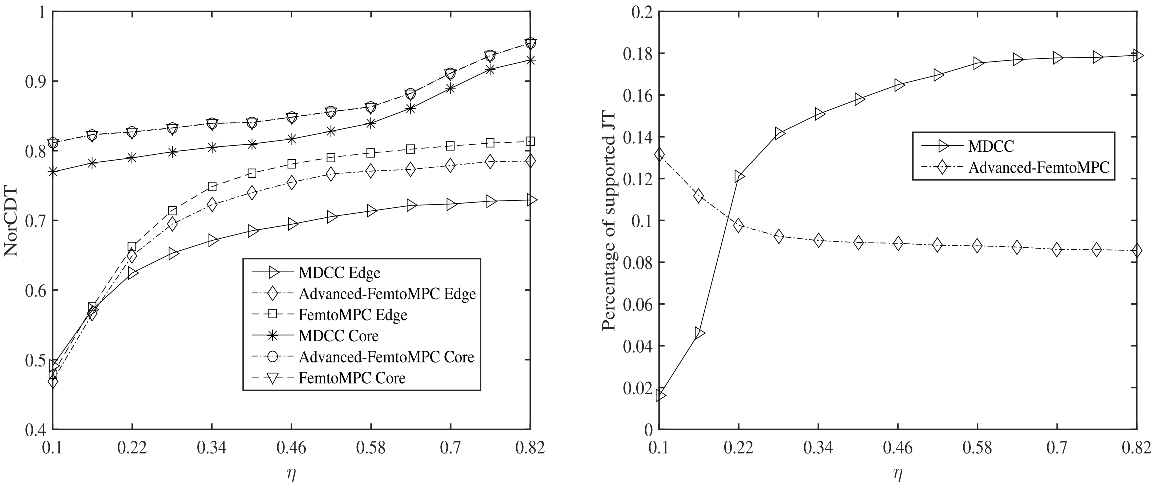
In order to demonstrate the influence of user request on the delay performance of cell-core MTs and cell-edge MTs. The NorCDT versus the user distribution ratio η is shown in Fig. 9(a). In this simulation, the MTs in each cell are in motion, and the user distribution ratio η varies from 0.1 to 0.8. We observe that the cell-core MTs experience lower NorCDT in the MDCC scheme than in the other schemes. Nevertheless, the cell-edge MTs in the MDCC scheme experience almost the same NorCDT as in the other schemes when η holds a relatively small value. This is because when η is small, there is a large number of MTs waiting to be served by ST. In this case, to improve the cache hit rate, the MDCC scheme prefers to cache more different files. However, as the number of requests from the cell-edge area continues to grow,



(a) NorCDT of cell-core MTs

(b) NorCDT of cell-edge MTs

Fig. 8. NorCDTs of cell-core MTs (a) and cell-edge MTs (b) respectively under different values of Zipf parameter α, where percentage of cache size $= 0.2$.



(a) NorCDT

(b) Percentage of supported JTs

Fig. 9. Influence of user distribution ratio η (the ratio of the number of cell-edge MTs to the number of cell-core MTs) on NorCDT (a) and percentage of supported JTs (b) for cell-core MTs and cell-edge MTs, where Zipf $\alpha = 0.7$, percentage of total cache size $= 0.2$.

the cell-edge MTs experience lower NorCDT in the MDCC scheme than in the other schemes and the performance gap between different schemes becomes larger. This is because the MDCC scheme stimulates the BSs to produce more JT opportunities for the cell-edge MTs when more requests are from the cell-edge area. This can be also explained by Fig. 9(b), which demonstrates the influence of user distribution η on the percentage of supported JTs for the MDCC and Advanced-FemtoMPC schemes. In the beginning, the percentage of supported JTs in the MDCC scheme is lower than that in the Advanced-FemtoMPC scheme. Then, the percentage of supported JTs goes up sharply in the MDCC scheme and drops slowly in the Advanced-FemtoMPC scheme as η increases, and it finally stays constant in both schemes. Meanwhile, the proposed scheme can support up to 17% of the total JTs, and outperforms the Advanced-FemtoMPC scheme with around 9% improvement for the entire system, which greatly improves the throughput of cell-edge MTs. This also reflects that the MDCC scheme can maintain good performance under different user distribution ratios, which clearly indicates the stability of our proposed scheme.

VI. Conclusion

In this paper, we explored the problem of caching in the CoMP-integrated UDCN, where the cache-equipped BSs can support ST and JT simultaneously to serve MTs. To minimize content delivery delay, we proposed the MDCC scheme. Compared with the existing caching schemes, our scheme exploits storage-dimension and transmission-dimension cooperations to jointly optimize the content placement for JT and ST. Based on the cooperation strategy, we formulated the file caching problem to be a NLIP problem and developed a GA-based algorithm to solve it. The simulation results show the impacts of cache capacity, content popularity, and request pattern on the NorCDT. They demonstrate that our scheme can reduce content delivery delay for both cell-core and cell-edge MTs.

More importantly, in our scheme, the file placement can dynamically adapt to the user distribution (i.e., the ratio of the number of cell-edge MTs to the number of cell-core MTs) in a cell, which has practical meaning for the deployment of caching in UDCNs.

APPENDIX

A. Proof for the Equivalence Between (13) and (14)

Given the optimal caching and routing decisions of Problem (I), $\boldsymbol{x}^* = \{(x_{b_i,f_k})^*\}_{b_i\in\mathcal{B},f_k\in\mathcal{F}}$ and $\boldsymbol{y}^* = \{(y^{f_k}_{b_i,b_r})^*\}_{b_i\in\mathcal{B},b_r\in\mathcal{N}_{b_i},f_k\in\mathcal{F}}$, under constraints $c4$ and $c5$, proving the equivalence between (13) and (14) is equivalent to confirming that the optimal results $(x_{b_i,f_k})^*$ and $\left(y^{f_k}_{b_i,f_k}\right)^*$ satisfy

$$(x_{b_i,f_k})^* + \sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* = \bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_r,f_k})^*, \tag{23}$$

where $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_i,f_k})^* \in \{0,1\}$ and $\sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* \in \{0,1\}$ based on constraint $c5$. Then, we discuss the results from the following two cases.

Case 1: For a certain (b_i, f_k) making $(x_{b_i,f_k})^* = 1$, we can obtain $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_i,f_k})^* = 1$. Furthermore, based on constraint $c5$, we obtain $\sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* \leq 1-(x_{b_i,f_k})^* = 0$. As a result, we obtain $\sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* = 0$, thus (23) holds.

Case 2: For a certain (b_i, f_k) making $(x_{b_i,f_k})^* = 0$, based on constraint $c5$, we have $\sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* \leq 1$. Moreover, we have $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_r,f_k})^* \in \{0,1\}$. To further prove that (23) holds, we discuss the result $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_r,f_k})^*$ from two aspects:

- Assuming that $(x_{b_r,f_k})^* = 0$ for $\forall b_r \in \mathcal{N}_{b_i}$, we have $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_r,f_k})^* = 0$. Based on constraint $c4$, we obtain $\left(y^{f_k}_{b_i,b_r}\right)^* \leq (x_{b_r,f_k})^* = 0$ for $\forall b_r \in \mathcal{N}_{b_i}$. As a result, we have $\sum_{b_r\in\mathcal{N}_{b_i}} \left(y^{f_k}_{b_i,b_r}\right)^* \leq \sum_{b_r\in\mathcal{N}_{b_i}} (x_{b_r,f_k})^* = 0$. Thus, (23) holds.
- Otherwise, if $(x_{b_r,f_k})^* = 1, \exists b_r \in \mathcal{N}_{b_i}$, i.e., $\bigcup_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_r,f_k})^* = 1$, we first assume $\left(y^{f_k}_{b_i,b_r}\right)^* = 0$ for $\forall b_r \in \mathcal{N}_{b_i}$, then we will find another opportunity for the neighboring BS which has cached file f_k to transmit it to BS b_i. This will save the hackhaul transmit delay $d^{\mathrm{CP}}_{b_i}$ and lead to a lower objective value of Problem (II), which is contradictory to the precondition that $\left(y^{f_k}_{b_i,b_r}\right)^*$ is the optimal solution. So, we have $\left(y^{f_k}_{b_i,b_r}\right)^* = 1$, $\exists b_r \in \mathcal{N}_{b_i}$. Thus, (23) also holds.

Based on above analysis, the proof for (23) is complete. ■

B. Proof of Lemma 1

Obviously, $\bigcap_{b_r\in\mathcal{N}_{b_i}\cup\{b_i\}} (x_{b_i,f_k})^* \in \{0,1\}$. Similarly to the above proof, given the optimal results $\boldsymbol{x}^* = \{(x_{b_i,f_k})^*\}_{b_r\in\mathcal{B},f_k\in\mathcal{F}}$ and $\boldsymbol{z}^* = \{(z_{b_i,f_k})^*\}_{b_i\in\mathcal{B},f_k\in\mathcal{F}}$, we discuss the equivalence of (16) from two cases as follows:

Case 1: For a certain (b_i, f_k), if $(z_{b_i,f_k})^* = 1$, based on constraint $c3$, we obtain $(z_{b_r,f_k})^* = 1$ for $\forall b_r \in \mathcal{A}_{m_j}$. In addition, based on constraint $c2$, we have $(x_{b_r,f_k})^* \geq (z_{b_r,f_k})^* = 1$, thus we obtain $(x_{b_r,f_k})^* = 1$ for $\forall b_r \in \mathcal{A}_{m_j}$. As a result, we have $\bigcap_{b_r\in\mathcal{A}_{m_j}} (x_{b_r,f_k})^* = 1$. Thus (16) holds.

Case 2: For a certain (b_i, f_k), if $(z_{b_i,f_k})^* = 0$, based on constraint $c3$, we have $(z_{b_r,f_k})^* = 0$, for $\forall b_r \in \mathcal{A}_{m_j}$. Then, based on constraint $c2$, we have $(z_{b_r,f_k})^* \leq (x_{b_r,f_k})^* \in \{0,1\}$, for $\forall b_r \in \mathcal{A}_{m_j}$. To further prove that (16) holds, we discuss the results $(x_{b_r,f_k})^*$ from two aspects:

- Assuming that $(x_{b_r,f_k})^* = 0$, $\exists b_r \in \mathcal{A}_{m_j}$, we obtain $\bigcap_{b_r\in\mathcal{A}_{m_j}} (z_{b_i,f_k})^* = 0$. Thus $(x_{b_i,f_k})^* = \bigcap_{b_r\in\mathcal{A}_{m_j}} (x_{b_r,f_k})^* = 0$ holds.
- Otherwise, i.e., $(x_{b_r,f_k})^* = 1, \forall b_r \in \mathcal{A}_{m_j}$, we obtain $\bigcap_{b_r\in\mathcal{A}_{m_j}} (x_{b_r,f_k})^* = 1$, which produces an additional opportunity for the BSs in set $\mathcal{A}_{m_j}$ to perform JT for file f_k. This will lead to a lower objective value of Problem (I), which is contradictory with the precondition that $(z_{b_i,f_k})^*$ is the local or global optimal solution. Therefore, $\exists b_r \in \mathcal{A}_{m_j}$, for $(x_{b_r,f_k})^* = 0$. Thus (16) holds.

Based on the above analysis, the proof for Lemma 1 is complete. ■

REFERENCES

[1] 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.

[2] Cisco. (Mar. 2017). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Updata, 2016–2021*. [Online]. Available: https://www.cisco.com/

[3] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[4] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, "Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1207–1221, May 2016.

[5] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 15–21, Sep. 2016.

[6] S. H. Chae, T. Q. S. Quek, and W. Choi, "Content placement for wireless cooperative caching helpers: A tradeoff between cooperative gain and content diversity gain," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6795–6807, Oct. 2017.

[7] B. Zhou, Y. Cui, and M. Tao, "Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6284–6297, Sep. 2016.

[8] E. Demarchou, C. Psomas, and I. Krikidis, "Hybrid wireless edge caching for relaying with spatial randomness," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.

[9] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.

[10] B. Blaszczyszyn and A. Giovanidis, “Optimal geographic caching in cellular networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3358–3363.

[11] M. Ji, G. Caire, and A. F. Molisch, “The throughput-outage tradeoff of wireless one-hop caching networks,” *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, Oct. 2015.

[12] M. Taghizadeh, K. Micinski, C. Ofria, E. Torng, and S. Biswas, “Distributed cooperative caching in social wireless networks,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1037–1053, Jun. 2013.

[13] X. Li, X. Wang, K. Li, Z. Han, and V. C. M. Leung, “Collaborative multi-tier caching in heterogeneous networks: Modeling, analysis, and design,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6926–6939, Oct. 2017.

[14] P. Lin, Q. Song, Y. Yu, and A. Jamalipour, “Extensive cooperative caching in D2D integrated cellular networks,” *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2101–2104, Sep. 2017.

[15] 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.

[16] J. Ma, J. Wang, and P. Fan, “A cooperation-based caching scheme for heterogeneous networks,” *IEEE Access*, vol. 5, pp. 15013–15020, 2017.

[17] J. Lee *et al.*, “Coordinated multipoint transmission and reception in LTE-advanced systems,” *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 44–50, Nov. 2012.

[18] V. Jungnickel *et al.*, “The role of small cells, coordinated multipoint, and massive MIMO in 5G,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 44–51, May 2014.

[19] A. Liu and V. K. N. Lau, “Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems,” *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[20] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, “Cooperative caching and transmission design in cluster-centric small cell networks,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3401–3415, May 2017.

[21] F. Zhou, L. Fan, N. Wang, G. Luo, J. Tang, and W. Chen, “A cache-aided communication scheme for downlink coordinated multipoint transmission,” *IEEE Access*, vol. 6, pp. 1416–1427, 2018.

[22] A. Tuholukova, G. Neglia, and T. Spyropoulos, “Optimal cache allocation for femto helpers with joint transmission capabilities,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[23] C. Yang, H. Li, L. Wang, and Z. Xu, “A game theoretical framework for improving the quality of service in cooperative RAN caching,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[24] N. Rajatheva, *5G Mobile and Wireless Communications Technology*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[25] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, “Distributed caching for data dissemination in the downlink of heterogeneous networks,” *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.

[26] S. H. Chae and W. Choi, “Caching placement in stochastic wireless caching helper networks: Channel selection diversity via caching,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6626–6637, Oct. 2016.

[27] C. Giovanna, G. Massimo, and M. Luca, “On the performance of bandwidth and storage sharing in information-centric networks,” *Comput. Netw.*, vol. 57, pp. 3743–3758, Dec. 2013.

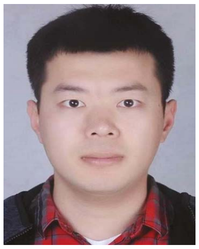
[28] K. Sastry, D. E. Goldberg, and G. Kendall, “Genetic algorithms,” in *Search Methodologies*. Boston, MA, USA: Springer, 2014.

[29] J. Jia, Y. Deng, J. Chen, A. H. Aghvami, and A. Nallanathan, “Achieving high availability in heterogeneous cellular networks via spectrum aggregation,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10156–10169, Nov. 2017.

[30] S. Lee, S. K. Lee, K. Kim, and Y. H. Kim, “Base station placement algorithm for large-scale LTE heterogeneous networks,” *PLoS ONE*, vol. 10, Oct. 2015, Art. no. e0139190.

[31] Z. Li and G. Simon, “In a Telco-CDN, pushing content makes sense,” *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 3, pp. 300–311, Sep. 2013.

[32] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, “Using partially overlapping channels to improve throughput in wireless mesh networks,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1720–1733, Nov. 2012.



Peng Lin received the M.S. degree in communication and information systems from Northeastern University, Shenyang, China, in 2017. He is currently pursuing the Ph.D. degree with Northeastern University and the Chongqing University of Post and Telecommunications, Chongqing, China. He is also a Visiting Student with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. His current research interests include mobile edge caching, edge computing, and machine learning in wireless networks.

Qingyang Song received the Ph.D. degree in telecommunications engineering from the University of Sydney, Australia. She is currently a Professor with the Chongqing University of Post and Telecommunications, Chongqing, China, and with the School of Computer Science and Engineering, Northeastern University, Shenyang, China. She has authored more than 80 articles in major journals and international conferences. Her current research interests are in radio resource management, cognitive radio networks, cooperative communications, ad hoc networks, mobile caching, and wireless powering.

Abbas Jamalipour (S’86–M’91–SM’00–F’07) received the Ph.D. degree in electrical engineering from Nagoya University, Nagoya, Japan. He is currently a Professor of ubiquitous mobile networking with the University of Sydney, Australia. He has authored 9 technical books, 11 book chapters, more than 450 technical articles, and five patents, all in the area of wireless communications. He is a fellow with the Institute of Electrical, Information, and Communication Engineers (IEICE) and the Institution of Engineers Australia, and an ACM Professional Member. He is an Elected Member of the Board of Governors and the President of the IEEE Vehicular Technology Society. He was the Editor-in-Chief of the IEEE WIRELESS COMMUNICATIONS, the Vice President-Conferences and a member of Board of Governors of the IEEE Communications Society. He has been an editor for several journals and is on the Editorial Board of IEEE ACCESS. He was a recipient of a number of prestigious awards such as the 2016 IEEE ComSoc Distinguished Technical Achievement Award in Communications Switching and Routing, the 2010 IEEE ComSoc Harold Sobol Award, the 2006 IEEE ComSoc Best Tutorial Paper Award, and 15 best paper awards. He has been a General Chair or a Technical Program Chair of a number of conferences, including IEEE ICC, GLOBECOM, WCNC, and PIMRC. He is an IEEE Distinguished Lecturer.