IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 15, 2021, accepted February 24, 2021, date of publication March 10, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065017

Extensive Cooperative Content Caching and Delivery Scheme Based on Multicast for D2D-Enabled HetNets

QINXUE FU, LIANXIN YANG, BAOQUAN YU, AND YAN WU
College of Communication Engineering, Army Engineering University of PLA, Nanjing 210007, China

Corresponding author: Qinxue Fu (fuqinxue99@163.com)

This work was supported in part by the Jiangsu Provincial Natural Science Fund for Outstanding Young Scholars under Grant BK20180028, in part by the Natural Science Foundations of China under Grant 61671474, and in part by the Jiangsu Provincial Natural Science Fund for Excellent Young Scholars under Grant BK20170089.

ABSTRACT The 5th generation (5G) mobile communication system demands low delay and high user rate of experience for mobile data services, which contradicts with the explosive growth of global mobile data traffic. The explosive growth of data traffic brings serious burden of backhaul links and deteriorates the performance of backhaul delay, and the performance of backhaul delay determines the user rate of experience and average downloading delay of users to a great extent. Cooperative content caching and delivery technique in edge caching entities is an effective method to solve the above problems. This paper proposes an extensive cooperative content caching and delivery scheme based on multicast for device-to-device(D2D)-enabled heterogeneous cellular networks (HetNets). We firstly design an extensive cooperative content caching (EC^3) scheme for D2D-enabled HetNets, which does not only consider the cooperative caching among D2D user, small base station (SBS) and macro base station (MBS) levels, but also considers the cooperative caching within each level. Moreover, our proposed EC^3 scheme also considers the influence of the backhaul links and remote server caching for cooperative caching. By introducing a content request probability (CRP) for each user predicted by context information, the EC^3 is formulated as an integer linear programming (ILP) problem. A hybrid genetic algorithm (HGA) is proposed for solving the problem, which combines genetic algorithm (GA), simulated annealing algorithm (SA) and local content request probability priority algorithm (LCRPPA). Furthermore, we design a scheme of content delivery based on multicast (CDBM), which is solved by a suboptimal edge priority algorithm based on multicast (EPABM). Simulation results show that the proposed extensive cooperative content caching and delivery scheme based on multicast can significantly improve the system performance compared with the existing cooperative content caching and delivery schemes.

INDEX TERMS Extensive cooperative content caching, low delay, multicast, content request probability (CRP).

I. INTRODUCTION

With the large-scale application of mobile terminals and the continuous emergence of new application services, the global mobile data traffic appears explosive growth. According to Cisco's [1] forecast, mobile data traffic will reach 587 EB in 2021, which will increase to 122 times compared with that of ten years ago. The explosive growth of global mobile data traffic gives birth to the 5th generation (5G) mobile communication system, and brings two challenges to existing mobile communication system: first, limited wireless bandwidth is difficult to adapt to the exponential growth of mobile data traffic; besides, the explosive growth of mobile data traffic has seriously aggravated backhaul burden [2], [3], which increases the total backhaul delay of content items from remote servers to base station(BS) and reduces user rate of experience. 5G mobile communication system uses network heterogeneity and edge caching to solve the above problems [4]–[7].

Small base stations (SBSs) (microcells, picocells, femtocell, etc) can be introduced in a cellular network with a macro base station (MBS) to form a heterogeneous network.

The associate editor coordinating the review of this manuscript and approving it for publication was Cunhua Pan.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Also, SBSs can use spatial frequency reuse technique to increase bandwidth utilization efficiency per unit area, that is, two SBSs far away can use the same bandwidth resources. In order to alleviate backhaul burden, device-to-device (D2D) communication is introduced in 5G heterogeneous cellular networks (HetNets). D2D communication enables content items to be transmitted directly between adjacent users without forwarding through MBS [8].

Most of the rapid increasing data traffic is mainly generated by the action that users repeatedly copy a small number of popular content items from remote servers through backhaul links [9]. To avoid redundant data transmission, edge caching technique is introduced in HetNets. The content items that users are interested in can be directly cached in the edge caching entities close to the users, which can satisfy users' request without fetching content items from remote servers. In D2D-enabled HetNets, user equipment's, SBSs and MBS can all cache content items, thus requesting users can be served by the above edge caching entities [10]. If none of the above caching entities cache the content items that requesting users demand, the requesting users have to fetch the content items from the remote Internet servers by accessing the backhaul links (e.g., optical fiber).

Since the mobility of users, the dynamics of content and the constraint of the caching capacity in caching entities, the caching hit rate of cached content items may not be high [9]. Therefore, designing an efficient content caching scheme is the key to improve the caching efficiency of caching room for D2D-enabled HetNets. There are two problems need to be solved: where to cache and what to cache [11]? For the first problem, content items can be cached in remote servers connected with the core networks and edge caching entities. For the second problem, we need to accurately predict the content request probability (CRP) of each user for all content items. Most of the existing researches assume that each user in HetNets has the same CRP, and the popularity of content items is used as the global content request probability (GCRP) instead of each user's CRP to introduce the content caching scheme [12]–[17]. The popularity of content items in most existing works is modeled as ZipF distribution; the probability that each user requests content item c_i is defined as $p_i = (1/i^{\gamma})/\sum_{l=1}^{F} 1/l^{\gamma}$, where γ denotes shape parameter [9], [12]–[17]. Since the popularity of content items cannot reflect the preference of user, the caching hit rate and caching efficiency of content items may not be high, especially when many requesting users' interests and preferences differ greatly. Therefore, accurately to predict the CRP of each user for all content items is important for an efficient caching scheme. The content caching schemes of [6] and [7] consider requesting users' interests and preferences for different content items in D2D-enabled caching cellular networks and effectively reduce the average content downloading delay and traffic load. However, the above works cannot consider the cooperative caching between edge caching entities, which limits further improvements in caching efficiency of content items.

In order to further improve caching efficiency of content items and eliminate redundant caching in edge caching entities, it is necessary to extensively consider the cooperation among edge caching entities in HetNets. In recent years, cooperative caching for HetNets has attracted the widespread interests of much research [2], [9], [17]–[21]. According to different optimization objectives, the existing caching schemes for cooperative caching in HetNets can be divided into three categories: probability (such as caching hit rate or successful transmission probability) [17], [18], energy efficiency [2] and bandwidth utilization efficiency (such as transmission rate or delay) [9], [19], and [20]. In [17], the author proposes a cooperative caching scheme between SBSs and D2D users, which transforms the content caching problem for HetNets into an integer linear programming (ILP) problem to maximize the local caching hit rate. However, due to the assumption that SBSs and users have the same coverage radius and power, its application scope is greatly limited. The author in [18] studies the cooperative content caching problem between BS and D2D users in D2D-based HetNets, which formulates the optimal caching scheme through maximizing the successful transmission probability. The author of [2] attributes cooperative content caching problem between MBS level and SBS level to the minimization of the average energy consumption. In [2], SBS group and MBS cooperate to jointly determine the caching scheme. [9], [19] and [20] take the average downloading delay as the objectives to optimize the cooperative caching schemes. The author in [9] proposes an optimal cooperative content caching and delivery scheme for HetNets. The caching scheme considers the mutual cooperation between the FBSs and D2D users, and assumes that MBS cache all the content items, from which the requesting users can obtain all content items. By applying queuing and optimization theory, a multi-BS cooperative caching scheme is proposed in [19]. In [20], the author proposes a cooperative caching scheme for D2D-enabled HetNets, which includes single-tier (i.e., BS tier or user tier) cooperation and cross-tier cooperation between BSs and user equipments.

Few of all above works extensively consider the cooperative caching among MBS level, SBS level and D2D user level, and this may lead to the reduction of caching hit rate and caching efficiency. Besides, most of above works assume that all content items that users demand can be fetched from MBS, which ignores the constraint of MBS' caching capacity and the influence of backhaul links and remote server caching for caching scheme. The reduction of caching hit rate and the neglect for MBS's caching capacity make a large number of content items be fetched from remote servers via MBS, which further aggravates the backhaul burden and results in a sharp increase of backhaul delay. Therefore, the average downloading delay increases sharply and the user rate of experience decrease seriously.

In this paper, we design an extensive cooperative content caching and delivery scheme based on multicast to solve the contradiction between explosive growth of mobile traffic and

the demand for low delay and high user rate of experience in 5G HetNets. The main contributions of this paper are as follows.

- Firstly, we propose an extensive cooperative content caching scheme for D2D-enabled HetNets. The scheme does not only consider the cooperative caching among D2D user, SBS and MBS levels, but also the cooperative caching within each level. Besides, we consider influence of backhaul links and remote server caching for the proposed caching scheme.
- Secondly, a CRP predicted by context information is introduced to transform the extensive cooperative content caching problem into an ILP problem, and we design a HGA to solve the problem.
- Thirdly, we design a content delivery scheme based on multicast. A suboptimal EPABM is proposed to solve the content delivery problem.

The rest of this paper is organized as follows. In Section II, we describe the system model. An extensive cooperative content caching problem for HetNets is formulated, and we propose a HGA to solve the problem in Section III. Section IV proposes a content delivery scheme based on multicast, which is solved by EPABM. In Section V, simulation results are used to evaluate the performance of our proposed extensive cooperative content caching and delivery scheme based on multicast. Finally, our work is concluded in Section VI. The main symbols and variables used in this paper are summarized in Table 1.

II. SYSTEM MODEL

As is shown in Fig. 1, we consider a D2D-enabled HetNet with a single-cell, which consists of a MBS, N SBSs and K users. There is a MBS located at the center of the whole area, and SBSs and users are randomly distributed. The whole area is covered by the MBS, in which each SBS only covers a small part of region and adjacent SBSs may overlap with each other. Signaling and data can be transmitted between

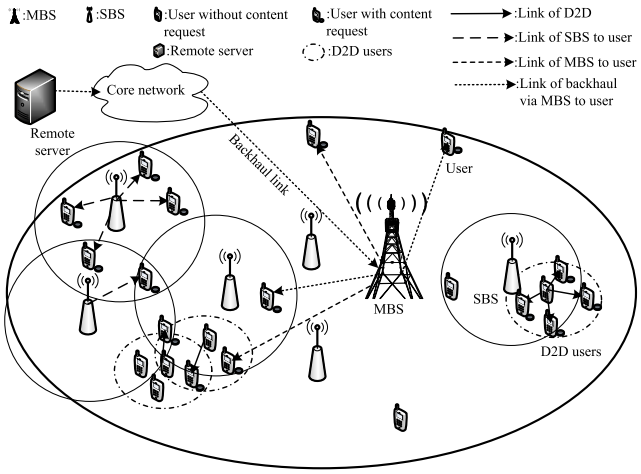


FIGURE 1. D2D-enabled HetNet system architecture for content caching and delivery.

TABLE 1. Main symbols and variables list.

Parameters	Description
p_i^k	Probability that u_k requests c_i
d_{kj}	Distance between u_k and a_j
R_{kj}	Achievable downloading rate from a_j to u_k
D_{kj}	Downloading delay from a_j to u_k
X	The caching scheme
Y	The transmission scheme
$\mathscr{H}_U(k)$	Users that can be connected with u_k
$\mathscr{H}_S(k)$	SBSs that can be connected with u_k
$\mathscr{H}_M(k)$	MBS that can be connected with u_k
$\mathscr{H}_0(k)$	All caching entities that can be connected with u_k
$\mathscr{I}_C^0(k)$	Content items none of which have been cached in caching entities of $\mathscr{H}_0(k)$
$\mathscr{I}_C(k)$	Content items each of which has been cached in more than a caching entity of $\mathscr{H}_0(k)$
$\mathbf{M}^{con}$	Global matrix that a_j can connect with u_k
$\mathscr{C}_M^0$	Content items that can be cached in free capacity of MBS
$\mathscr{C}_S^j$	Content items that can be cached in free capacity of SBS a_j
$\mathscr{P}_M$	Define a local average CRP set based on content set $\mathscr{C}_M^0$ in the coverage of MBS
$\mathscr{P}_S^j$	Define a local average CRP set $\mathscr{P}_S^j$ based on content set $\mathscr{C}_S^j$ in the coverage of SBS a_j
η_{best}^G	The chromosome that has the largest fitness value
φ_{best}^G	The largest fitness value corresponding to η_{best}^G
K	Number of users
N	Number of SBSs
F	Number of content items
C_U	The caching capacity of each user
C_S	The caching capacity of each SBS
C_M	The caching capacity of MBS
B_E	The sub-channel bandwidth
L	The size of each content item
D_B	The backhaul delay
N_{gro}^G	The population size in GA
N_{gen}	The evolution generation size in GA
N_{sel}^G	Number of selected chromosome in GA
p_{mut1}^G	The mutation probability in GA
p_{mut2}^G	Gene mutation probability in GA
τ_{ki}	Requesting variable whether u_k requests c_i
$\mathbf{X}_{opt}$	Caching scheme obtained from HGA
$\mathbf{V}$	Achievable downloading rate matrix
N^R	Number of content requests
U^R	Requesting user vector
C^R	Demanded content item vector
T^R	Content transport type vector
A^R	Content delivery entity vector
N_{mul1}	Multicast cluster head vector
V_{mul1}	Multicast threshold rate vector
S^{cha}	Available sub-channel number of caching entities
$\mathbf{Z}$	Delivery decision matrix for caching entities
$\mathbf{Z}_B$	Delivery decision vector for backhaul
$A_d^R(t)$	The t-th content delivery entity
C_d^R	Demanded content items cached in MBS
U_t^R	Requesting users whose requesting content items are cached in $A_d^R(t)$
C_t^R	Content items that requesting users of U_t^R demand
b_S	Number of sub-channels of each SBS
b_M	Number of sub-channels of MBS
$R_{\min}$	The minimum downloading rate

adjacent two SBSs through wired or wireless links [2]. The coverage radiuses of the MBS and each SBS are r_M and r_S, respectively. r_U is the maximum distance that arbitrary two D2D users can be connected with each other. The signal

transmission power of the MBS, SBSs and D2D user equipments is respectively P_M, P_S and P_U. The MBS is connected with the core network by high-capacity backhaul links (e.g., optical fiber), which is connected with remote servers cached all content items [22]. We denote the sets of users, MBS and SBSs as $\mathscr{U} = \{u_1, u_2, \ldots, u_K\}$, $\mathscr{M} = \{m_b\}$ and $\mathscr{S} = \{s_1, s_2, \ldots, s_N\}$ respectively. The union of user set, MBS set and SBS set is defined as caching entity set $\mathscr{A} = \mathscr{U} \cup \mathscr{M} \cup \mathscr{S} = \{a_1, a_2, \ldots, a_{K+N+1}\}$ for convenience of notation, i.e., $\{a_1, a_2, \ldots, a_K\} = \{u_1, u_2, \ldots, u_K\}$, $\{a_{K+1}\} = \{m_b\}$ and $\{a_{K+2}, a_{K+3}, \ldots, a_{K+N+1}\} = \{s_1, s_2, \ldots, s_N\}$.

$\mathscr{H}_U(k) = \{a_j \,|\, \|u_k - a_j\| \le r_U, a_j \in \mathscr{U}, j \in \{1, 2, \ldots, K\}\}$ denotes all users that can be connected with u_k through D2D links.$\mathscr{H}_S(k) = \{a_j \,|\, \|u_k - a_j\| \le r_S, a_j \in \mathscr{S}, j \in \{K+2, K+3, \ldots, K+N+1\}$ denotes all SBSs that can be connected with u_k. We define $\mathscr{H}_M(k) = \{a_{K+1}\}$ as the set which represents MBS that can be connected with u_k. For convenience, we define $\mathscr{H}_0(k) = \mathscr{H}_U(k) \cup \mathscr{H}_M(k) \cup \mathscr{H}_S(k)$ as the set of all caching entities that user u_k can be connected with.

Users can request content items from a finite content library set, which is denoted as $\mathscr{C} = \{c_1, c_2, \ldots, c_F\}$. Each content item is assumed with the same size of L bits, which can be removed when the content item is divided into blocks of the same length [23]. Each of MBS, SBSs and users can cache some content items for a long time and is assumed to equip with a local caching capacity of C_M, C_S and C_U respectively. The MBS, SBSs and users can fetch some popular content items that requesting users may demand from remote servers during off-peak times or periods of low traffic load, e.g. nighttime [17]. We introduce some incentive mechanisms, such as refunding schemes in literature [24] to encourage caching-enabled users to provide content service for other users by D2D links. A user can fetch a content item from all edge caching entities that can be connected with the user if the content item is cached in these caching entities. When a user cannot fetch a content item from edge caching entities, it must fetch the content item from remote servers.

We assume that MBS caches the location information, the caching information and the channel state information (CSI) of MBS, SBSs and user equipments, and the MBS can allocate resources for all communication links [11], [24]. Time is assumed to be slotted in our system, and we research the system in a specific time period T_0 [9]. In order to avoid interferences between the MBS and the SBSs, the MBS and SBSs need to be allocated different bandwidth resources. Bandwidth resources of the MBS and each SBS are equally divided into b_M and b_S sub-channels for MBS-user links and SBS-user links within the time period T_0 according to $b_M = \lfloor B_M^{\max}/B_M \rfloor$ and $b_S = \lfloor B_S^{\max}/B_S \rfloor$($\lfloor \cdot \rfloor$ represents the largest integer not more than "$\cdot$"), where $B_M^{\max}$, B_M, $B_S^{\max}$ and B_S represent the maximum available bandwidth capacity of MBS, the sub-channel bandwidth capacity of MBS, the maximum available bandwidth capacity of each SBS and the sub-channel bandwidth capacity of each SBS respectively [25]. We assume that the sub-channels of directly adjacent two SBSs are orthogonal so that SBS cellular links in two SBSs far away can reuse the same sub-channels, which can greatly increase available bandwidth capacity of SBSs. MBS is assumed to dynamically allocate sub-channels of different bandwidth resources to D2D users by using some complex resources allocation scheme, so as to avoid interferences between D2D links and cellular links [26]. We define the sub-channel bandwidth of each D2D pair as B_U. In this paper, we assume that D2D link, SBS-user link and MBS-user link have the same sub-channel bandwidth B_E, which stands for each sub-channel bandwidth of caching entities.

Through above assumption, main interferences between caching entities are eliminated. Thus we can ignore the interferences and use signal to noise ratio (SNR) to substitute for signal to interference plus noise ratio (SINR) when we compute the achievable downloading rate (bit/s). The achievable downloading rate from $a_j \in \mathscr{H}_0(k)$ to u_k can be denoted as $R_{kj} = B_E \log_2(1 + SNR_{kj})$, and $R_{kj} = 0$ when $a_j \notin \mathscr{H}_0(k)$, where SNR_{kj} denotes the SNR of the signal transmitted from caching entity a_j to user u_k. We assume P_j is denoted as the transmission power of a_j, thus SNR_{kj} can be defined as $SNR_{kj} = \frac{P_j h_{kj} d_{kj}^{-\alpha}}{B_E N_0}$, where d_{kj}, h_{kj}, $d_{kj}^{-\alpha}$, α and N_0 respectively represent the distance from a_j to u_k, channel gain of small-scale Rayleigh fading from a_j to u_k, channel gain of large-scale fading from a_j to u_k, path loss exponent and the unilateral power spectral density of the additive white Gaussian noise. The random variable h_{kj} obeys an exponential distribution with average power of unity, i.e., $h \sim \exp(1)$[27]. The downloading delay D_{kj} corresponding to the achievable downloading rate R_{kj} is defined as $D_{kj} = \frac{L}{R_{kj}}$, where L denotes the size of each content item.

Users' content request arrivals are modeled as independently Poisson processes with average request arrival rate λ_k (arrival/time period) for an arbitrary user u_k [28]. In order to achieve reasonable optimization of edge caching, user CRP need to be accurately estimated. Considering the importance of each user's individual interest, personal background and social ties for user CRP, we use a prediction model proposed in [17] to directly give CRP of user u_k for content item c_i with p_i^k, which combines the social network with recommendation algorithms based on context-aware and collaborative filtering (CF). Thus request arrival rate that user u_k requests content item c_i is $\lambda_{ki} = \lambda_k \cdot p_i^k$.

The total bandwidth capacity allocated for all delivered content items which are cached in local caching capacity of a caching entity must satisfy the constraint of bandwidth capacity of the caching entity in the time period T_0; otherwise the content items that cannot satisfy the constraint can only be fetched from remote servers. We assume that MBS has a temporary register equipped with some caching capacity. When users request content items from remote servers, the remote servers first transmit the content items to MBS and make them temporarily cached in the temporary register, and then the content items are forwarded from the temporary register to the requesting users. When the links by which the content items cached in the temporary register of MBS

can be transmitted are allocated enough bandwidth capacity, the above content items can be immediately delivered from MBS to requesting users; otherwise the content items need postponing delivery for several time periods (generally speaking, the delay that the content items are postponed delivery is one to several time periods in the worst case when the backhaul burden is not greatly heave and the network has not been congested) until there is enough bandwidth capacity that can be allocated to the above links. In this paper, the total backhaul delay from remote servers to MBS is modeled as an exponentially distributed random variable with a given mean value D_B, and we replace the backhaul delay with D_B for simplicity [25]. In general, the average backhaul delay is much greater than one time period. Therefore, we cannot allocate bandwidth resources for delivering the content items that are cached in remote servers in the time period T_0, and can only allocate bandwidth resources for the delivery of the above content items in some time periods after the demanded content items have been transmitted to MBS and cached in the temporary register of MBS. In some time period after T_0, we use complex resource allocation scheme [11], [24] to allocate remaining bandwidth resources of caching entities for delivering several demanded content items that have been cached in the temporary register of MBS, which are determined to be fetched from remote servers in the time period T_0 according to some delivery scheme. Through several time periods after T_0, we assume that all demanded content items can be completely delivered to the requesting user according to the above way, which are determined to be fetched from remote servers in the time period T_0. Since the delay from the time when the demanded content items are cached in the temporary register of MBS to the time when the demanded content items start deliver from MBS is very low compared with average backhaul delay, we assume the delay is 0, which can be easily removed by adding a small average delay to the backhaul delay.

III. EXTENSIVE COOPERATIVE CONTENT CACHING SCHEME

In section III, we formulate the extensive cooperative content caching problem as an ILP problem, and then propose a HGA to find the suboptimal solution of the problem.

A. PROBLEM FORMULATION

We define a binary caching decision variable $x_{ij} \in \{0, 1\}$, which indicates whether content item c_i is cached in caching entity a_j or not: $x_{ij} = 1$ when c_i is cached in a_j and 0 otherwise. Then a binary caching decision matrix is introduced to describe the caching scheme, which is denoted as $\mathbf{X} = \{x_{ij} \,|\, a_j \in \mathscr{A}, c_i \in \mathscr{C}\}$ [22]. The problem of extensive cooperative content caching can be considered as how to cache the content items at user equipments, SBSs, MBS in order to minimize the average downloading delay of users for given constraints. The average downloading delay of users does not only depend on users' content requests, but it also depends on caching entities which the content items are delivered from. Therefore, we need to consider caching and delivery jointly in extensive cooperative content caching scheme. A binary transmission decision variable is defined as $y_{kj} \in \{0, 1\}$, which denotes whether caching entity $a_j \in \mathscr{A}$ can be selected to directly transmit a content item to requesting user u_k or not: $y_{kj} = 1$ when a content item is selected and 0 otherwise. A binary transmission decision matrix is introduced to describe the transmission scheme, which is denoted as $\mathbf{Y} = \{y_{kj} \,|\, u_k \in \mathscr{U}, a_j \in \mathscr{A}\}$ [25]. The largest number of content requests that caching entity a_j can serve simultaneously, which is denoted as B_j, depends on the number of sub-channels that a_j can provide. Therefore, B_j can be derived as follows [9]:

$$B_j = \begin{cases} 1 - \sum_{i=1}^{F} \lambda_j p_i^j (1 - x_{ij}), & \mathrm{a}_j \in \mathscr{U} \\ \left\lfloor dB_S^{\max} B_S \right\rfloor, & \mathrm{a}_j \in \mathscr{S} \\ \left\lfloor \dfrac{B_M^{\max}}{B_M} \right\rfloor, & \mathrm{a}_j \in \mathscr{M}, \end{cases} \tag{1}$$

where $\sum_{i=1}^{F} \lambda_j p_i^j (1 - x_{ij})$ represents the probability that user $\mathrm{a}_j \in \mathscr{U}$ has a content request that need to be served in the time period T_0. We assume an arbitrary user a_j is either a requester or a server at the same time period, so the largest number of content requests that a_j can serve is $1 - \sum_{i=1}^{F} \lambda_j p_i^j (1 - x_{ij})$. When a requesting user u_k fetches a content item from different kinds of edge caching entities or remote server, there will be different downloading delay. We assume that D_{avg}^E and D_{avg}^B denote the average downloading delay of users when content items are fetched from edge caching entities and remote servers, respectively. Therefore, the total average downloading delay of users D_{avg} can be denoted as equality (2). D_{avg}^B consists of the last two items in (2), which denote the two kinds of downloading delay of users when content items are fetched from remote servers via MBS. The two kinds of downloading delay correspond to two cases, respectively. The first case is that a requesting user has no choice but to fetch the content item from a remote server since all edge caching entities that the requesting user can be connected with have not cached the content item. The next is that though edge caching entities that a requesting user can be connected with cache the content item, the caching entities have no available sub-channels to transmit the content item.

$$\begin{aligned} D_{avg} &= D_{avg}^E + D_{avg}^B \\ &= \sum_{i=1}^{F} \sum_{a_j \in \mathscr{H}_0(k)} \sum_{k=1}^{K} \lambda_k p_i^k x_{ij} y_{kj} D_{kj} \\ &\quad + \sum_{i=1}^{F} \sum_{k=1}^{K} \lambda_k p_i^k (1 - \sum_{a_j \in \mathscr{H}_0(k)} x_{ij})(D_B + D_{k,K+1}) \\ &\quad + \sum_{i=1}^{F} \sum_{a_j \in \mathscr{H}_0(k)} \sum_{k=1}^{K} \lambda_k p_i^k x_{ij} (1 - y_{kj})(D_B + D_{k,K+1}) \end{aligned} \tag{2}$$

The problem of optimal extensive cooperative content caching (denoted by EC^3) which minimizes the total average downloading delay of users is formulated as follows:

$$\min_{\mathbf{X},\mathbf{Y}} D_{avg}, \tag{3}$$

$$s.t. \sum_{i=1}^{F} x_{ij} L \leqslant \begin{cases} C_U, & \mathrm{a}_j \in \mathscr{U} \\ C_S, & \mathrm{a}_j \in \mathscr{S} \\ C_M, & \mathrm{a}_j \in \mathscr{M}, \end{cases} \tag{4}$$

$$\sum_{k=1}^{K} \lambda_k y_{kj} \leqslant B_j, a_j \in \mathscr{A}, \tag{5}$$

$$\sum_{j=1}^{K+N+1} y_{kj} \leqslant 1, \quad u_k \in \mathscr{U}, \tag{6}$$

$$y_{kj} \leqslant \max_{c_i \in \mathscr{C}} \{x_{ij}\}, \quad a_j \in \mathscr{H}_0(k), u_k \in \mathscr{U}, \tag{7}$$

$$\sum_{a_j \in \mathscr{H}_0(k)} x_{ij} \leqslant 1, \quad c_i \in \mathscr{C}, u_k \in \mathscr{U}. \tag{8}$$

Inequality (4) represents the caching capacity constraint of users, SBSs and MBS [2]. Inequality (5) indicates bandwidth capacity (sub-channel) constraint of users, SBSs and MBS [9]. Inequality (6) reveals that at most one in all caching entities can serve the requesting user in the HetNet at the same time [9]. Inequality (7) indicates that as long as there is a content item cached in edge caching entity a_j, a_j can be selected to transmit the content item to multiple requesting users. From inequality (7), we can see that a content item cached in a caching entity can be transmitted to multiple requesting users at the same time, which ensures the realization of the following GA and SA algorithms and is also conducive to the formation of multicast in following proposed content delivery scheme. Inequality (8) shows that content item $c_i \in \mathscr{C}$ can be cached in at most one edge caching entity which can be connected with u_k. Through (8), we can see MBS, SBSs, D2D users which can be connected with u_k cooperate to cache a content item for u_k and the content item is cached in at most one of caching entities, which effectively improves the utilization efficiency of the caching room. It is clear that the EC^3 problem is an ILP problem, which is NP-hard. We prove that the EC^3 problem is NP-hard in the following Lemma 1.

Lemma 1: The EC^3 problem is NP-hard.

Proof: The lemma can be proved through restriction [29]. We consider a special case of the EC^3 problem with $x_{ij} = 0, a_j \in \mathscr{U} \cup \mathscr{S}, \forall c_i \in \mathscr{C}; x_{ij} = 1, a_j \in \mathscr{M}, \forall c_i \in \mathscr{C}; \sum_{i=1}^{F} L \leqslant C_M, \mathrm{a}_j \in \mathscr{M}$. Thus constraints (4), (7) and (8) can be removed from the EC^3 problem. It's obvious that the above special case is a classic assignment problem. Since assignment problem is NP-hard [30], [31], the EC^3 problem is also NP-hard. ■

B. SOLUTION ALGORITHM

Since EC^3 problem is NP-hard, it is difficult to find the optimal solution. Moreover, to relax (7) is very difficult. We combine genetic algorithm (GA), simulated anneal algorithm (SA) and local content request probability priority algorithm (LCRPPA) to obtain a hybrid genetic algorithm (HGA) and use the HGA to find the suboptimal solution of the EC^3 problem [2], [32]-[35]. The framework of HGA is illustrated in Fig. 2.

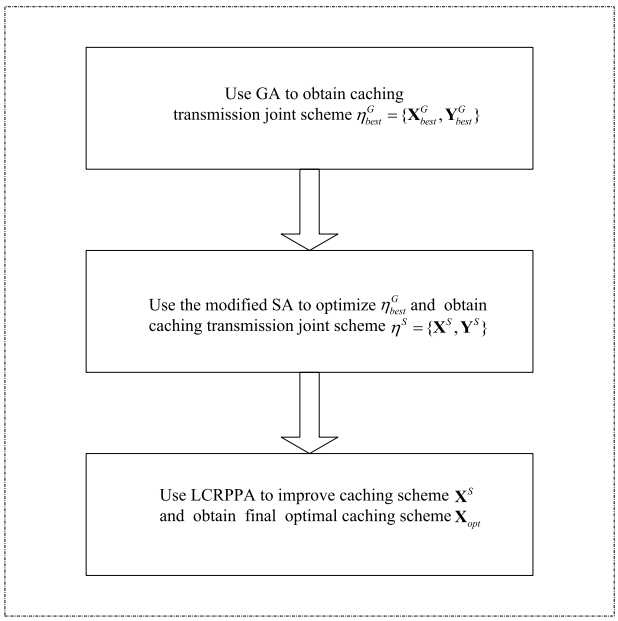


FIGURE 2. The framework of HGA for EC^3.

1) REVISION ALGORITHM

The number of available caching schemes and transmission schemes which are generated randomly or by heuristic algorithm may become tremendous with the rapid increase of the number of users and content items. However, since the constraints (8) and (6) in the EC^3 scheme, most of the above schemes have to discarded, which lead to a serious increase of computation. So we need to find a revision algorithm (RA) to modify the above schemes to make them satisfy the above constraints of EC^3. The major role of RA is to make the caching schemes and transmission schemes generated randomly or by heuristic algorithm respectively satisfy constraints (8) and (6) through modifying x_{ij}, which have satisfied other constraints of EC^3 except the constraints (8) and (6).

At the beginning of RA, input initial parameters. We assume that the caching scheme $\mathbf{X}$ and transmission scheme $\mathbf{Y}$ that input RA satisfy other constraints of EC^3 except the constraints (8) and (6). Initialize $\mathscr{U}' = \emptyset$, where $\mathscr{U}'$ represents a caching entity set whose caching scheme has been modified. After the k-th time iteration, all caching entities of $\mathscr{H}_0(k)$ join $\mathscr{U}'$. A caching entity cannot cache new content items after it joins $\mathscr{U}'$ (but content items cached in the caching entity can be removed from it). After the k-th time iteration, caching schemes of caching entities connected with user u_k satisfy the constraint (8). Through following K

times iterations, we modify caching scheme $\mathbf{X}$ and make the modified caching scheme satisfy the constraint (8).

In the k-th time iteration, we define three sets $\mathscr{H}'(k)$, $\mathscr{I}_C^0(k)$ and $\mathscr{I}_C(k)$. $\mathscr{H}'(k)$ denotes caching entities which can be connected with u_k and whose caching schemes can be modified in this iteration. $\mathscr{I}_C^0(k)$ denotes the content items none of which have been cached in caching entities of $\mathscr{H}_0(k)$. $\mathscr{I}_C(k)$ represents the content items each of which has been cached in more than a caching entity of $\mathscr{H}_0(k)$. Since content items of $\mathscr{I}_C(k)$ do not satisfy the constraint (8) for u_k, we need to remove all content items from $\mathscr{I}_C(k)$ until $\mathscr{I}_C(k) = \emptyset$.

Randomly select $c_i \in \mathscr{I}_C(k)$, and define two caching entity sets $\mathscr{H}_i'(k)$ and $\mathscr{H}_i(k)$, which represent caching entities that has been cached c_i and are in $\mathscr{H}'(k)$ and $\mathscr{H}_0(k)$ respectively. In caching entities of $\mathscr{H}_i(k)$, except the caching entity closest to u_k, c_i cached in other caching entities need to be replaced c_i with different content items in $\mathscr{I}_C^0(k)$, i.e., after c_i cached in above some caching entity is replaced with a content item in $\mathscr{I}_C^0(k)$, the content item need to be removed from $\mathscr{I}_C^0(k)$. Therefore, when the content item number of $\mathscr{I}_C^0(k)$ is less than the number of these caching entities that need be replaced with c_i, c_i in the above caching entities that have not been replaced direct is removed. Through the above process, only a caching entity can cache c_i in all caching entities which can be connected with u_k, which satisfies the constraint (8) for c_i. At this time, c_i is removed from $\mathscr{I}_C(k)$. Note that symbol $|\cdot|$ represents the number of elements in set "$\cdot$".

Repeat the above process until $\mathscr{I}_C(k) = \emptyset$. In this way, the modified caching schemes of caching entities connected with user u_k satisfy the constraint (8). Since caching schemes of caching entities in $\mathscr{H}_0(k)$ have been modified, so make them join $\mathscr{U}'$.

Through K times iterations, we obtain the modified caching scheme which satisfies the constraint (8) for all users. Then we modify transmission scheme $\mathbf{Y}$. Define two user sets $\mathscr{I}_U^0$ and $\mathscr{I}_U$. $\mathscr{I}_U^0$ denotes the users that none of caching entities can be selected to transmit a content item to. $\mathscr{I}_U$ represents the users that more than one caching entity can be selected to transmit a content item to. According to the constraint (6), a requesting user can be served by at most one caching entity. So we need to remove all caching entities from $\mathscr{I}_U$. Through the same way like removing all content items from $\mathscr{I}_C(k)$ in $\mathbf{X}$, we remove all requesting users from $\mathscr{I}_U$ and modify transmission scheme $\mathbf{Y}$. The modified transmission scheme satisfy the constraint (6). Output modified caching scheme and transmission scheme.

Algorithm 1 shows the revision algorithm (RA).

Algorithm 1 Revision Algorithm (RA)

Input: $\mathbf{X}$, $\mathbf{Y}$, r_U, r_S, r_M, the location of users, SBSs and MBS;

Output: $\mathbf{X}'$ and $\mathbf{Y}'$;

Define a caching entity set $\mathscr{U}'$, and initialize $\mathscr{U}' = \emptyset$;
for $k = 1 : K$
  Define a caching entity set $\mathscr{H}'(k) = \mathscr{H}_0(k) - \mathscr{U}' \cap \mathscr{H}_0(k)$, which denotes caching entities that can serve u_k and whose caching schemes can be modified in the k-th time iteration;
  Define two content sets $\mathscr{I}_C^0(k) = \{c_{i'} \Big| \sum_{a_j \in \mathscr{H}_0(k)} x_{i'j} = 0, \forall x_{i'j} \in \mathbf{X}, \forall c_{i'} \in \mathscr{C}\}$ and $\mathscr{I}_C(k) = \{c_i \Big| \sum_{a_j \in \mathscr{H}_0(k)} x_{ij} > 1, \forall x_{ij} \in \mathbf{X}, \forall c_i \in \mathscr{C}\}$, and $\mathscr{I}_C(k)$ denotes content items that cannot satisfy the constraint (8);
  while $\mathscr{I}_C(k) \neq \emptyset$
    Randomly select a content item $c_i \in \mathscr{I}_C(k)$ and define two caching entity sets $\mathscr{H}_i'(k) = \{a_{j'} \,|x_{ij'} = 1, \forall a_{j'} \in \mathscr{H}'(k), x_{ij'} \in \mathbf{X}\}$, $\mathscr{H}_i(k) = \{a_j \,|x_{ij} = 1, \forall a_j \in \mathscr{H}_0(k), \ x_{ij} \in \mathbf{X}\}$;
    while $(\mathscr{I}_C^0(k) \neq \emptyset) \& (|\mathscr{H}_i'(k)| > 1)$
      Select $a_{j'} \in \mathscr{H}_i'(k)$, and make $a_{j'}$ have the shortest distance from user u_k; then set $x_{ij'} = 0$;
      Randomly select $c_{i'} \in \mathscr{I}_C^0(k)$, then set $x_{i'j'} = 1$; remove $a_{j'}$ from $\mathscr{H}_i'(k)$ and $\mathscr{H}_i(k)$, and remove $c_{i'}$ from $\mathscr{I}_C^0(k)$.
    end while
    while $|\mathscr{H}_i(k)| > 1$
      Randomly select $a_{j_0'} \in \mathscr{H}_i(k)$, then set $x_{ij_0'} = 0$;
      remove $a_{j_0'}$ from $\mathscr{H}_i(k)$;
    end while
    Remove c_i from $\mathscr{I}_C(k)$;
  end while
  Update $\mathscr{U}' = \mathscr{U}' \cup \mathscr{H}_0(k)$;
end for
Define two user sets $\mathscr{I}_U^0 = \{u_{k'} \Big| \sum_{a_j \in \mathscr{A}} y_{k'j} = 0, \forall y_{k'j} \in \mathbf{Y}, \forall u_{k'} \in \mathscr{U}\}$ and $\mathscr{I}_U = \{u_k \Big| \sum_{a_j \in \mathscr{A}} y_{kj} > 1, \ \forall y_{kj} \in \mathbf{Y}, \forall u_k \in \mathscr{U}\}$, and $\mathscr{I}_U$ denotes requesting users that cannot satisfy the constraint (6);
Similar to the process that all content items in $\mathscr{I}_C(k)$ are removed, we remove all requesting users from $\mathscr{I}_U$ and modify transmission scheme $\mathbf{Y}$;output modified caching scheme $\mathbf{X}'$ and modified transmission scheme $\mathbf{Y}'$.

2) LOCAL CONTENT REQUEST PROBABILITY PRIORITY ALGORITHM

In the caching scheme obtained by heuristic algorithm, some caching entities may have some free caching capacity that has not yet been used because of constraint (8). In free caching capacity of a caching entity, we can cache some content items that have the highest local average CRP in the coverage of the caching entity by LCRPPA, which can effectively increase caching efficiency of caching room.

We input the global connection matrix $\mathbf{M}^{con}$, caching scheme $\mathbf{X}^S$ (obtained by SA) and other parameters. $\mathbf{M}^{con}$ is defined as $\mathbf{M}^{con} = \{m_{kj} | m_{kj} = 1, \forall u_k \in \mathscr{U}, \forall a_j \in \mathscr{H}_U(k) \cup \mathscr{H}_S(k) \cup \mathscr{H}_M(k); m_{kj} = 0 \text{ otherwise}\}$ where a binary variable $m_{kj} \in \{0, 1\}$ denotes whether caching entity $a_j \in \mathscr{A}$ can be connected with user $u_k \in \mathscr{U}$ or not:$m_{kj} = 1$ if a_j can be connected with user u_k; $m_{kj} = 0$ otherwise.

At the beginning of LCRPPA, we first make some feasible content items cached into free caching capacity of MBS. If MBS has free caching capacity, we select some content items from a feasible content set and make them cached into free caching capacity of MBS according to the local average CRP in the coverage of MBS, where the feasible content set consists of the content items that have not yet been cached in MBS and satisfy constraint (8) (i.e., the content items cannot be cached in caching entities which can be connected with users in the coverage of MBS). We define a content set $\mathscr{C}_M^0$, which denotes content items that have not been cached in MBS and satisfy constraint (8). $\mathscr{C}_M^0$ is a cacheable content set for free capacity of MBS. In the coverage of MBS, define a local average CRP set $\mathscr{P}_M$ based on content set $\mathscr{C}_M^0$. We find content items corresponding to the first $\min\{\left|\mathscr{C}_S^j\right|, C_S - \sum_{c_i \in \mathscr{C}} x_{ij}\}$ items of highest probability in $\mathscr{P}_M$ and cache the content items into free capacity of MBS.

We cache some feasible content items into free capacity of SBSs according to following way. For an arbitrary SBS a_j, we first define a user set $\mathscr{U}_S^j$ and a caching entity set $\mathscr{A}_S^j$, which respectively represent users that can be connected with a_j and caching entities which can be connected with all users in $\mathscr{U}_S^j$. A content set $\mathscr{I}_S^j$ denotes content items that all caching entities has cached, which can serve the users that a_j can be connected with. So $\mathscr{C}_S^j = \mathscr{C} \backslash \mathscr{I}_S^j$ indicates the content items that have not been cached in SBS a_j and satisfy constraint (8). So $\mathscr{C}_S^j$ become a cacheable content set for SBS a_j. Similar to MBS, we define a local average CRP set $\mathscr{P}_S^j$ for a_j based on $\mathscr{C}_S^j$. Then we find the content items corresponding to the first $\min\{\left|\mathscr{C}_S^j\right|, C_S - \sum_{c_i \in \mathscr{C}} x_{ij}\}$ items of highest probability in $\mathscr{P}_S^j$ and make the content items cached into free capacity of SBS a_j. In the same way, we cache some feasible content items into free capacity of all SBSs.

Through the same way as $a_j \in \mathscr{S}$, we cache some feasible content items into free capacity of all $a_j \in \mathscr{U}$. Since some users cannot be connected with any other users by D2D link, we can only cache the content items that have not yet been cached in themselves into their own free caching capacity according CRP of themselves, which makes the constraint (7) of EC^3 be satisfied. We modify caching scheme $\mathbf{X}^S$ and obtain modified caching scheme $\mathbf{X}_{opt}$.

Algorithm 2 displays the local content request probability priority algorithm (LCRPPA).

Algorithm 2 Local Content Request Probability Priority Algorithm (LCRPPA)

Input: $p_i^k, \lambda_k, \mathbf{X}^S, \mathbf{M}^{con}, C_U, C_S, C_M$;
Output: $\mathbf{X}_{opt}$;

if $\sum_{c_i \in \mathscr{C}} x_{ij} \neq C_M$, for $\forall x_{ij} \in \mathbf{X}^S, a_j \in \mathscr{M}$
  Define a content set $\mathscr{C}_M^0 = \{c_{i_0} \left| \sum_{a_j \in \mathscr{A}} x_{i_0 j} = 0, \forall c_{i_0} \in \mathscr{C}, \forall x_{i_0 j} \in \mathbf{X}^S\right.\}$, which denotes content items that can be cached in free capacity of MBS;
  In the coverage of MBS, define a local average CRP set based on content set $\mathscr{C}_M^0$ as $\mathscr{P}_M = \{p_{i_0} \left| p_{i_0} = \sum_{u_k \in \mathscr{U}} p_{i_0}^k \lambda_k / \sum_{u_k \in \mathscr{U}} \sum_{c_i \in \mathscr{C}} p_i^k \lambda_k, \forall c_{i_0} \in \mathscr{C}_M^0\right.\}$;
  Find the content items corresponding to the first $\min\{\left|\mathscr{C}_M^0\right|, C_M - \sum_{c_i \in \mathscr{C}} x_{i,K+1}\}$ items of highest probability in $\mathscr{P}_M$, and cache them into free capacity of MBS;
end if
for $j = K + 2 : K + N + 1$
  if $\sum_{c_i \in \mathscr{C}} x_{ij} \neq C_S$, for $\forall x_{ij} \in \mathbf{X}^S, a_j \in \mathscr{S}$
    Define a user set and a caching entity set as $\mathscr{U}_S^j = \{u_k | m_{kj} = 1, m_{kj} \in \mathbf{M}^{con}, \forall u_k \in \mathscr{U}, a_j \in \mathscr{S}\}$, $\mathscr{A}_S^j = \{a_{j_1} \left| m_{k_1 j_1} = 1, m_{k_1 j_1} \in \mathbf{M}^{con}, \forall u_{k_1} \in \mathscr{U}_S^j, \forall a_{j_1} \in \mathscr{A}\right.\}$;
    Define a content set $\mathscr{I}_S^j = \{c_i \left| x_{ij_1} = 1, \forall a_{j_1} \in \mathscr{A}_S^j, \forall c_i \in \mathscr{C}, \forall x_{ij_1} \in \mathbf{X}^S\right.\}$ and find a content set $\mathscr{C}_S^j = \mathscr{C} \backslash \mathscr{I}_S^j$, where $\mathscr{C}_S^j$ represents content items that can be cached into free capacity of SBS a_j;
    In the coverage of SBS a_j, define a local average CRP set $\mathscr{P}_S^j$ based on content set $\mathscr{C}_S^j$;find the content items corresponding to the first $\min\{\left|\mathscr{C}_S^j\right|, C_S - \sum_{c_i \in \mathscr{C}} x_{ij}\}$ items of highest probability in $\mathscr{P}_S^j$, and cache them into free capacity of SBS a_j;
  end if
end for
Through the same way as $a_j \in \mathscr{S}$, we cache the content items that have not yet been cached in a_j and satisfy constraint (8) into free caching capacity of $a_j \in \mathscr{U}$;
For some users which cannot be connected with any other users by D2D link, we cache content items that have not yet been cached in them into their free caching capacity according CRP of themselves;
We obtain modified caching scheme $\mathbf{X}_{opt}$.

3) HYBRID GENETIC ALGORITHM

SA is a random search algorithm to search the optimal solution by simulating solid annealing process. In the search process, the simulated annealing algorithm combines the probability jump characteristics to randomly find the global optimal solution of the objective function in the solution space, that is, it jumps out of the local optimum with a certain

probability and finally tends to the global optimum [32]. Compared with GA, SA has better local searching ability, however, it cannot cover all solution sets in solution space and may fall into local minimum [32]. GA is a kind of random search algorithm to search the optimal solution by simulating the natural evolution process, and it is also a kind of population algorithm [33]. Compared with SA, GA has strong global search ability and can obtain all solution sets in solution space [33], [34], however, its local searching ability is inferior to SA. Therefore, we must combine the advantages of both to design a better search algorithm.

Due to the better global searching ability of GA, we first use GA to search the global optimal solution of caching schemes; and then put the optimal solution obtained from GA into SA to search a superior solution to the above optimal solution, which is because of the better local searching ability of SA.

SA starts with an initial solution and current temperature $T^S = T^S_{\max}$ ($T^S_{\max}$ is the maximum temperature), randomly change the current solution in its neighborhood to generate a new solution, and then compute the objective function of the new solution. If the objective function is less than that of the previous solution, the new solution is accepted, otherwise rejected. When the feasible solution is updated iteratively, a new solution which is worse than the previous solution may be accepted with a certain probability. At the same temperature, the algorithm iterates N^S (repeated cooling times at the same temperature) times and T^S value is attenuated according to $T^S = \alpha^S T^S$ (α^S is cooling coefficient) until T^S value reaches $T^S_{\min}$(the minimum temperature). Note that in SA we compute the objective function according to (2). Since the SA may transfer from current solution to a worse solution with a certain probability, i.e., to lose the current superior solution, we can modify the SA in [2] by adding a storage step to save the past superior solution when the new solution will be accepted [32], then we obtain a modified SA. So we can combine GA, modified SA and LCRPPA to obtain a HGA scheme to solve EC^3 problem.

Input evolution generation size N^G_{gen}, population size N^G_{gro}, the number of chromosomes selected in HGA N^G_{sel}, mutation probability p^G_{mut1}, gene mutation probability p^G_{mut2} and other parameters.

At the beginning of HGA, we compute initial value d_{kj}, R_{kj}, D_{kj} and $\mathbf{M}^{con}$. GA analyzes the evolution of a chromosome group, so we first need to find some feasible chromosomes that have largest fitness value as an input of evolution. Initialize and modify a chromosome group and obtain η_{cur0}, which consists of N^G_{gro} caching transmission joint scheme. Each caching transmission joint scheme consists of a caching scheme and a transmission scheme. Compute the fitness value of each chromosome according to $\varphi(\cdot) = 1/D_{avg}(\cdot)$, and select first N^G_{sel} items of chromosomes that have the largest fitness value in η_{cur0} to form chromosome group η_{cur}, which is used as input chromosome group in the following crossover operators. Set the chromosome that has the largest fitness value in η_{cur} as η^G_{best}, whose corresponding fitness value $\varphi(\eta^G_{best})$ is set as the largest fitness value φ^G_{best}. We perform the evolution of a chromosome group η_{cur} in GA through the following N^G_{gen} times iterations.

When each chromosome in the previous generation is given, the chromosome group in the next generation can be generated by two candidate operators: crossover and mutation [2]. ''Parent'' chromosomes can use crossover and mutation to obtain ''child'' chromosomes.

We use crossover operator to obtain ''child'' chromosomes and save the optimal solution. Input η_{cur} as the chromosome group in the current generation. Randomly arrange chromosomes in η_{cur} and put these chromosomes into a chromosome pool. Select direct adjacent two chromosomes in turn as two ''parent'' chromosomes from the pool, and select a crossover point from $\{F, 2F, \ldots, NF\}$[2]. The genes after two ''parent'' chromosomes' crossover points are exchanged with each other and use RA to modify the two chromosomes to generate two new ''child'' chromosomes. Note that the exchange are executed respectively according to the caching scheme and the transmission scheme. Add ''child'' chromosomes obtained from crossover to ''parent'' chromosomes and form a chromosome group η_1, then compute the fitness value of each chromosome in η_1. Select first N^G_{sel} items of chromosomes that have the largest fitness value in η_1 to form chromosome group η_2. Afterwards, find chromosome η^G_{best2} corresponding to the largest fitness value in η_2, and denote the largest fitness value as $\varphi(\eta^G_{best2})$. If $\varphi^G_{best} < \varphi(\eta^G_{best2})$, respectively save η^G_{best2} and $\varphi(\eta^G_{best2})$ as η^G_{best} and φ^G_{best}.

We use mutation operator to obtain ''child'' chromosomes and save the optimal solution. Select the n-th chromosome in η_2 with a low probability p^G_{mut1}, then randomly select $(N + K + 1)p^G_{mut2}$ columns of $\eta^{(n)}_2$ to perform mutation. Use RA to modify the chromosome obtained from mutation and generate a new chromosome $\eta^{(n)}_4$. Inherit $\eta^{(n)}_4$ to the next generation chromosome $\eta^{(n)}_{new}$. If mutation is not performed, directly inherit $\eta^{(n)}_2$ to $\eta^{(n)}_{new}$. Select in turn all chromosomes in η_2 to perform mutation and we obtain a new chromosome group $\eta_{new} = \{\eta^{(n)}_{new}\}, n = 1, 2, \ldots, N^G_{sel}$. Find chromosome η^G_{best3} corresponding to the largest fitness value in η_{new}, and denote the largest fitness value as $\varphi(\eta^G_{best3})$.If $\varphi^G_{best} < \varphi(\eta^G_{best3})$, respectively save η^G_{best3} and $\varphi(\eta^G_{best3})$ as η^G_{best} and φ^G_{best}. We update $\eta_{cur} = \eta_{new}$ after this evolution.

Through above N^G_{gen} times iterations, we obtain the chromosome with the highest fitness value $\eta^G_{best} = \{\mathbf{X}^G_{best}, \mathbf{Y}^G_{best}\}$ as a superior solution in GA. Then we use the modified SA to optimize η^G_{best} and obtain $\eta^S = \{\mathbf{X}^S, \mathbf{Y}^S\}$. Afterward, we use LCRPPA to improve caching scheme $\mathbf{X}^S$ and obtain $\mathbf{X}_{opt}$ as a suboptimal solution.

Algorithm 3 shows the hybrid genetic algorithm (HGA).

After triple iteration, in the worst case, the complexity of the fitness function and RA are all $O(F \times K \times (N + K + 1))$. Therefore, the computational complexity of GA in HGA is $O(N^G_{gen} \times N^G_{sel} \times F \times K \times (N + K + 1))$, and the complexity

Algorithm 3 Hybrid Genetic Algorithm (HGA)

Input: $K, N, F, T^S_{\min}, T^S_{\max}, \alpha^S, N^S, N^G_{gen}, N^G_{gro}, N^G_{sel}, p^G_{mut1}, p^G_{mut2}, p^k_i, \lambda_k, L, C_U, C_S, C_M, b_M, b_S, B_E, P_U, P_S, P_M, r_U, r_S, r_M, N_0, \alpha, h_{kj}, D_B$, the location of users,SBSs and MBS;

Output: $\mathbf{X}_{opt}, R_{kj}, \mathbf{M}^{con}$;

Compute d_{kj}, R_{kj} and D_{kj}, for $a_j \in \mathscr{A}$, $u_k \in \mathscr{U}$; then find $\mathbf{M}^{con}$ defined as Algorithm 2;
Initialize a chromosome group η_0, and the n_g-th caching transmission joint scheme is $\eta_0^{(n_g)} = (\mathbf{X}^{(n_g)}, \mathbf{Y}^{(n_g)}), n_g = 1, 2, \ldots, N^G_{gro}$, where $\mathbf{X}^{(n_g)}$ and $\mathbf{Y}^{(n_g)}$ must satisfy the constraints (4) and (5), respectively;
Use RA to modify $\eta_0^{(n_g)}$ and obtain a new scheme $\eta_{cur0}^{(n_g)} = (\mathbf{X}_{cur0}^{(n_g)}, \mathbf{Y}_{cur0}^{(n_g)})$ so that $\mathbf{X}_{cur0}^{(n_g)}$ and $\mathbf{Y}_{cur0}^{(n_g)}$ respectively satisfy the constraints (8) and (6), $n_g = 1, 2, \ldots, N^G_{gro}$;
Compute the fitness value $\varphi(\eta_{cur0}^{(n_g)})$ of each chromosome in η_{cur0} according to $\varphi(\eta_{cur0}^{(n_g)}) = 1/D_{avg}(\eta_{cur0}^{(n_g)})$, where $\eta_{cur0} = \{\eta_{cur0}^{(n_g)}\}, n_g = 1, 2, \ldots, N^G_{gro}$;
Select first N^G_{sel} items of chromosomes that have the largest fitness value in η_{cur0} to form chromosome group η_{cur}; set the chromosome that has the largest fitness value in η_{cur} as η^G_{best}, whose corresponding fitness value $\varphi(\eta^G_{best})$ is set as the largest fitness value φ^G_{best};
for $g = 1 : N^G_{gen}$
  Randomly arrange chromosomes in η_{cur} and put these chromosomes into a chromosome pool;
  for $n = 1 : N^G_{sel}$
    Select the n-th and $(n+1)$-th chromosomes as two "parent" chromosome from pool, and randomly select a crossover point from $\{F, 2F, \ldots, NF\}$;
    The parts after the two "parent" chromosomes' crossover points are exchanged with each other and then modify the two chromosomes by RA to obtain two "child" chromosomes;
  end for
  Add "child" chromosomes to "parent" chromosomes and form a chromosome group η_1, where the n_c-th chromosome $\eta_1^{(n_c)} = (\mathbf{X}_1^{(n_c)}, \mathbf{Y}_1^{(n_c)})$, $n_c = 1, 2, \ldots, 3N^G_{sel}$;
  Compute the fitness value of chromosome $\eta_1^{(n_c)}$, $\varphi(\eta_1^{(n_c)}) = 1/D_{avg}(\eta_1^{(n_c)}), n_c = 1, 2, \ldots, 3N^G_{sel}$;
  Select first N^G_{sel} items of chromosomes that have the largest fitness value in η_1 to form chromosome group η_2, then find chromosome $\eta^G_{best2} = (\mathbf{X}^G_{best2}, \mathbf{Y}^G_{best2})$ corresponding to the largest fitness value in η_2 and its fitness value $\varphi(\eta^G_{best2})$;
  if $\varphi^G_{best} < \varphi(\eta^G_{best2})$
    Set $\eta^G_{best} = \eta^G_{best2}$ and $\varphi^G_{best} = \varphi(\eta^G_{best2})$;
  end if
  for $n = 1 : N^G_{sel}$
    if $c_1^{rand} = \text{rand}[0, 1] < p^G_{mut1}$
      Select the n-th chromosome $\eta_2^{(n)}$ in η_2 and further randomly select $(N + K + 1)p^G_{mut2}$ columns of $\eta_2^{(n)}$ to mutate $\eta_2^{(n)}$, then obtain chromosome $\eta_3^{(n)}$;
      Use RA to modify $\eta_3^{(n)}$ and obtain chromosome $\eta_4^{(n)}$, then inherit $\eta_4^{(n)}$ to the next generation chromosome $\eta_{new}^{(n)}$;
    else Directly inherit $\eta_2^{(n)}$ to $\eta_{new}^{(n)}$;
    end if
  end for
  Compute all fitness values of chromosomes in η_{new} ($\eta_{new} = \{\eta_{new}^{(n)}\}$, $n = 1, 2, \ldots, N^G_{sel}$), and find chromosome η^G_{best3} corresponding to the largest fitness value in η_{new}, and denote the largest fitness value as $\varphi(\eta^G_{best3})$;
  if $\varphi^G_{best} < \varphi(\eta^G_{best3})$
    Set $\eta^G_{best} = \eta^G_{best3}$ and $\varphi^G_{best} = \varphi(\eta^G_{best3})$;
  end if
  Update $\eta^{(n)}_{cur} = \eta^{(n)}_{new}, n = 1, 2, \ldots, N^G_{sel}$;
end for
The GA end, and obtain $\eta^G_{best} = \{\mathbf{X}^G_{best}, \mathbf{Y}^G_{best}\}$;
Use the modified SA to optimize η^G_{best}, and obtain $\eta^S = \{X^S, Y^S\}$;
Use LCRPPA to improve caching scheme $\mathbf{X}^S$, and obtain $\mathbf{X}_{opt}$.

of the SA and the LCRPPA are $O(\left\lfloor \log_{\alpha^S} \frac{T^S_{\min}}{T^S_{\max}} \right\rfloor \times N^S \times F \times K \times (N+K+1))$ and $O(K \times C_U + N \times C_S + C_M)$, respectively. So the computational complexity of the HGA for the EC^3 problem is given by $O(\max(\left\lfloor \log_{\alpha^S} \frac{T^S_{\min}}{T^S_{\max}} \right\rfloor \times N^S, N^G_{gen} \times N^G_{sel}) \times F \times K \times (N + K + 1))$.

IV. CONTENT DELIVERY SCHEME BASED ON MULTICAST

In order to improve bandwidth utilization efficiency and increase the average downloading rate experienced by users, we introduce multicast in content delivery scheme. We formulate the content delivery based on multicast (denoted by CDBM) as an ILP problem, and then use EPABM to solve it.

A. PROBLEM FORMULATION

In this paper, a user is assumed to request at most a content item at a time period. In a specific time period T_0, K different users may be generate N^R content requests, and these content requests can be the same or different. The process of generating content requests is assumed to be a stochastic process, and the statistics of the stochastic process are identical with those in EC^3 scheme of Section III [9]. Define a binary requesting variable $\tau_{ki} \in \{0, 1\}$ that represents whether user u_k has requested content item c_i, where $\tau_{ki} = 1$ indicates user u_k has requested content item c_i, $\tau_{ki} = 0$ otherwise. Since in the time period T_0 user u_k can request at most a content item,

τ_{ki} satisfies $\sum_{c_i \in \mathscr{C}} \tau_{ki} \in \{0, 1\}$. In this paper, u_k can become a requester only when u_k has not cached the content item that u_k demands. This is because we assume that a user is either a content provider or a content requester in the time period T_0[9]. So the probability of $\tau_{ki} = 1$ can be given by $\lambda_k p_i^k (1 - x_{ik}^*)$, where x_{ik}^* is the proposed EC^3 scheme.

In order to describe the CDBM problem, we introduce a binary delivery decision matrix $\mathbf{Z}$ and a binary delivery decision vector $\mathbf{Z}_B$. Define $\mathbf{Z}$ as $\mathbf{Z} = \{z_{kj} | u_k \in \mathscr{U}, a_j \in \mathscr{A}\}$, where the binary variable $z_{kj} \in \{0, 1\}$ represents whether a_j is selected to directly deliver a content item to u_k or not: $z_{kj} = 1$ if a_j is selected, $z_{kj} = 0$ otherwise [19]. Define $\mathbf{Z}_B$ as $\mathbf{Z}_B = \{z_k | u_k \in \mathscr{U}\}$, where the binary variable $z_k \in \{0, 1\}$ denotes whether the remote server is selected via MBS to deliver a content item to u_k: $z_k = 1$ if the remote server is selected, $z_k = 0$ otherwise. Then we denote the sum downloading rate of all users as:

$$R_{sum} = \sum_{u_k \in \mathscr{U}} \sum_{c_i \in \mathscr{C}} \tau_{ki} [\sum_{a_j \in \mathscr{H}_0(k)} z_{kj} x_{ij}^* R_{kj} + z_k \frac{L}{L/R_{k,K+1} + D_B}] \tag{9}$$

We maximize the sum downloading rate of all users, and formulate the content delivery scheme as follows:

$$\max_{Z, Z_B} R_{sum}, \tag{10}$$

$$s.t. \sum_{a_j \in \mathscr{H}_0(k)} z_{kj} + z_k = \sum_{c_i \in \mathscr{C}} \tau_{ki}, \quad u_k \in \mathscr{U}, \tag{11}$$

$$\sum_{u_k \in \mathscr{U}} z_{kj} \leqslant 1 - \sum_{c_i \in \mathscr{C}} \tau_{ji}, \quad a_j \in \mathscr{U}, \tag{12}$$

$$\sum_{u_k \in \mathscr{U}} z_{kj} \leqslant b_S, \quad a_j \in \mathscr{S}, \tag{13}$$

$$\sum_{u_k \in \mathscr{U}} z_{kj} \leqslant b_M, \quad \forall a_j \in \mathscr{M}, \tag{14}$$

$$z_{kj} R_{\min} \leqslant R_{kj}, a_j \in \mathscr{H}_0(k), \quad u_k \in \mathscr{U}. \tag{15}$$

Constraint (11) denotes that a content request must be served by a caching entity or remote server. Constraint (12) represents that $a_j \in \mathscr{U}$ is selected to serve u_k only when a_j does not generate a content request, and inequality (13) is the constraint of the available sub-channel number of each SBS [9]. Inequality (14) is the constraint of the available sub-channel number of MBS in the time period T_0. Inequality (15) indicates the constraint of the downloading rate at which a user can download a content item directly from an edge caching entity, where $R_{\min}$ represents the minimum downloading rate from a caching entity to a user. We can see that inequality (15) ensure users rate of experience at edge caching entity. It is clear that the above problem is an ILP problem, which is NP-hard. We prove that the above problem is NP-hard through considering a special case as the proof of in Lemma 1, which is that all content items have been cached in Misuser u_k fetch all demanded content item from MBS, and we won't repeat it here for the proof.

When multiple users request the same content item cached in the same edge caching entity, the edge caching entity may not serve for the users due to the constraint of bandwidth capacity; though the caching entity can deliver the content item to all requesting users by multiple sub-channels, bandwidth utilization is not efficient. In order to solve the above problem, we introduce multicast [21] in the above content delivery scheme to formulate a scheme of content delivery based on multicast (CDBM). We assume all edge caching entities can multicast a content item in the time period T_0. When multiple users need to fetch the same content item from the same edge caching entity, which the users can be connected with, we can use multicast to deliver the content item in order to improve bandwidth utilization efficiency. Multicast is controlled by MBS. A caching entity that multicasts a content item and all the requesting users that request the content item compose a multicast group, where the caching entity is called cluster head, and requesting users are called cluster members. The data downloading rate from cluster head to cluster members is the same in multicast. In order to ensure that each cluster member in the multicast group can all download the content item, the data downloading rate from cluster head to cluster members cannot be more than a maximum downloading rate, which equals the minimum rate of achievable downloading rate from cluster head to all cluster members in a multicast group. Since unicast is a special case of multicast, CDBM problem is also NP-hard.

B. SOLUTION ALGORITHM

Since the CDBM problem is NP-hard, it is difficult to obtain the global optimal solution. We propose a suboptimal edge priority algorithm based on multicast (EPABM) to solve the CDBM problem. The EPABM is essentially a greedy algorithm, the key of which is to make as many requesting users as possible fetch content items from edge caching entities (not from remote servers). We introduce multicast in EPABM. We first use edge priority algorithm regardless of bandwidth capacity (EPARBC) to determine the multicast content delivery scheme, and then use MABUS and MABM to modify it under the constraint of bandwidth capacity.

1) EDGE PRIORITY ALGORITHM REGARDLESS OF BANDWIDTH CAPACITY

A requesting user first chooses to fetch a content item from the unique one of caching entities that the requesting user can be connected with. When edge caching entities have not cached the content item or have no enough bandwidth capacity, the user must fetch the content item from the remote server via MBS. We first determine the content delivery scheme regardless of the constraint of bandwidth capacity.

We input the achievable downloading rate matrix $\mathbf{V} = \{R_{kj}\}$,caching scheme $\mathbf{X}_{opt}$ obtained from the above HGA and other parameters.

At the beginning of EPARBC, define row vectors U^R, C^R, T^R and A^R as requesting user vector, demanded content item vector, content transport type vector and content delivery

entity vector corresponding to requesting user set. $U^R(r)$ represents the user in $\mathscr{U}$ that corresponds to the r-th requesting user, e.g., $U^R(r) = k$ represents that the r-th requesting user is identical with u_k in $\mathscr{U}$. $C^R(r)$ represents the content item that the r-th requesting user demands, e.g., $C^R(r) = i$ represents the content item that the r-th requesting user demands is c_i. $T^R(r)$ and $A^R(r)$ represent content transport type and content delivery entity corresponding to the r-th requesting user respectively. $T^R(r) = 1$ indicates that the content item is delivered directly from the caching entity to the r-th requesting user; $T^R(r) = 2$ indicates that the content item is delivered from the remote server via MBS to the r-th requesting user. Determine U^R and C^R according to $\{\tau_{ki}\}$. We can determine the r-th requesting user's transport type $T^R(r)$, content delivery entity $A^R(r)$ through the following r-th time iteration.

For the r-th time iteration, first define a set with at most one element H_r^{dir}, which represents direct caching entity that the r-th requesting user can directly fetch the content item from. If $H_r^{dir} \neq \emptyset$, we can directly fetch a content item from H_r^{dir}, so set $T^R(r) = 1$ and $A^R(r) = H_r^{dir}(1)$; otherwise, the content item can be fetched from the remote server via MBS, and we set $T^R(r) = 2$ and $A^R(r) = K + 1$. If the achievable downloading rate from the content delivery entity (H_r^{dir}) to the r-th requesting user cannot satisfy the constraint of the minimum downloading rate, the user fetch a content item from the remote server and modify his T^R and A^R value. Through the above N^R times iterations, we obtain T^R and A^R without considering the constraint of bandwidth capacity.

Algorithm 4 shows the edge priority algorithm regardless of bandwidth capacity (EPARBC).

Algorithm 4 Edge Priority Algorithm Regardless of Bandwidth Capacity (EPARBC)

Input: $\mathbf{X}_{opt}, N^R, K, N, F, \tau_{ki}, \mathbf{M}^{con}, \mathbf{V} = \{R_{kj}\}$;
Output: U^R, C^R, T^R, A^R;

1: Define N^R dimensional row vectors U^R, C^R, T^R and A^R as requesting user vector, demanded content item vector, content transport type vector and content delivery entity vector, then initialize U^R, C^R, T^R and A^R;
2: Sort the set $\{\tau_{ki} \,|\tau_{ki} = 1, \forall u_k \in \mathscr{U}, \forall c_i \in \mathscr{C}\}$ in a ascend order according to k. If the r-th binary requesting variable of the above set that has been re-sorted is $\tau_{k_0 i_0}$, we obtain $U^R(r) = k_0$, and $C^R(r) = i_0$ for $\forall k_0 \in \{1, 2, \ldots, K\}, \forall i_0 \in \{1, 2, \ldots, F\}$, $\forall r \in \{1, 2, \ldots, N^R\}$;
3: **for** $r = 1 : N^R$
4: Define one set with at most one element $H_r^{dir} = \left\{j \,\middle|\, \mathbf{M}^{con}(U^R(r), j) = 1, \mathbf{X}_{opt}(C^R(r), j) = 1, \forall a_j \in \mathscr{A}\right\}$
5: **if** $H_r^{dir} \neq \emptyset$, then set $T^R(r) = 1$ and $A^R(r) = H_r^{dir}$;
6: **else** Set $T^R(r) = 2$ and $A^R(r) = K + 1$;
7: **end if**
8: **if** $T^R(r) \neq 2$
9: When the achievable downloading rate from the content delivery entity (H_r^{dir}) to the r-th requesting user is less than $R_{\min}$, set $T^R(r) = 2$ and $A^R(r) = K + 1$;
10: **end if**
11: **end for**

2) MULTICAST ALGORITHM BASED ON USERS AND SBSS

We use MABUS to modify the content delivery scheme. Input initial parameters. Define row vectors N_{mul1}, V_{mul1} and S^{cha}, then initialize them. N_{mul1} and V_{mul1} represent multicast cluster head vector and multicast threshold rate vector corresponding to requesting users when the content item is multicasted from caching entity cached the content item. S^{cha} represents available sub-channel number of caching entities. $N_{mul1}(r) = j(r \in \{1, 2, \ldots, N^R\}, a_j \in \mathscr{A})$ represents a content item is delivered from content delivery entity a_j to the r-th requesting user by multicast.$V_{mul1}(r)$ represents the multicast's max threshold rate value when a content item is delivered from a_j to the r-th requesting user by multicast. $S^{cha}(j)$ represents the number of available sub-channels of $a_j \in \mathscr{A}$ at present, which have not still been allocated.

Afterward, find a row vector $A_d^R = \text{unique}(A^R)$, where unique$(\cdot)$ represents the vector which consists of the different elements in "$\cdot$". A_d^R denotes all edge content delivery entities regardless of channel capacity. For the demanded content items cached in an arbitrary content delivery entity of A_d^R, they are delivered from the larger to the smaller according to their corresponding the number of requesting users until the available sub-channel number of the content delivery entity become 0. Note that the available sub-channel number of the content delivery entity need to be reduced by 1 after a content item cached in the content delivery entity is determined to be delivered. Through the following $\left|A_d^R\right|$ times iterations, we find out T^R, A^R, N_{mul1} and V_{mul1}.

Through the t-th time iteration, we obtain delivery scheme of content items cached in $A_d^R(t)$. Define two row vectors U_t^R and C_t^R, which represent all requesting users whose requesting content items are cached in $A_d^R(t)$ and the content items that requesting users of U_t^R demand, respectively. Find a row vector $C_{t_0}^R = \text{unique}(C_t^R)$. Sort all content items in $C_{t_0}^R$ in a descend order according to the number of different content items in C_t^R and obtain $C_{t'}^R$. We define the content item $C_{t'}^R(1)$ as $c_{t\max}^R$ which has the largest demanded times. Define two row vectors $U_{t\max}^R$ and U_{tmom}^R, which represent requesting users in U_t^R that demand the content item with the largest demanded times and requesting users in U_t^R that demand the other content items except $c_{t\max}^R$, respectively.

If $A_d^R(t) < K + 1$, $A_d^R(t)$ is a D2D user. We can find N_{mul1}, N_{mul2}, V_{mul1} and V_{mul2}, when multicast is performed at $A_d^R(t)$. Modify T^R and A^R according to the following case. If $\left|C_t^R\right| = 1$, there is only a content item in C_t^R that can be delivered from $A_d^R(t)$. The delivery scheme of content item cached in $A_d^R(t)$ can be obtained from Case A. If $\left|C_t^R\right| \neq 1$, there are multiple content items that need to be delivered. Because $A_d^R(t)$ has only one sub-channel, we can only deliver content item $c_{t\max}^R$ in $A_d^R(t)$. The delivery scheme

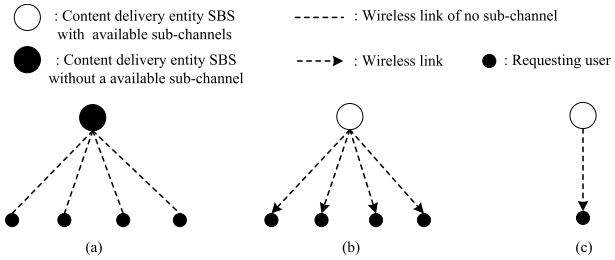


FIGURE 3. EPABM algorithm sketch diagram for content delivery.

of content items cached in $A_d^R(t)$ can be obtained from Case B. If $A_d^R(t) > K+1$, $A_d^R(t)$ is a SBS. Since a SBS has b_S sub-channel, the demanded content items that are cached in $A_d^R(t)$ should be delivered according to the number of their corresponding requesting users. The first b_S content items that are cached in $A_d^R(t)$ and correspond to the most requesting users are delivered. We decide on the delivery scheme of the content items of $C_{t'}^R$ through $\left|C_{t'}^R\right|$ time iteration as follows.

In the l-th time iteration, we firstly define one requesting user vector U_{tl}^R which represents the requesting users in U_t^R that request content item $C_{t'}^R(l)$. Then the delivery scheme of the content item $C_{t'}^R(l)$ is divided into two cases: $S^{cha}(A_d^R(t)) = 0$ and $S^{cha}(A_d^R(t)) \neq 0$. When $S^{cha}(A_d^R(t)) = 0$, i.e., available sub-channel number of SBS $A_d^R(t)$ is 0, requesting users in U_{tl}^R fetch content items from remote servers. The delivery scheme of $C_{t'}^R(l)$ can be obtained from Case C, which is shown in Fig. 3 (a). When $S^{cha}(A_d^R(t)) \neq 0$, $C_{t'}^R(l)$ can be delivered from $A_d^R(t)$ to requesting users in U_{tl}^R by unicast or multicast. When $\left|U_{tl}^R\right| = 1$, $C_{t'}^R(l)$ is directly delivered to U_{tl}^R by unicast according to Case D, which is shown in Fig. 3 (c). When $\left|U_{tl}^R\right| > 1$, $C_{t'}^R(l)$ can be delivered from $A_d^R(t)$ to requesting users in U_{tl}^R by multicast according to Case E, which is shown in Fig. 3 (b). Subtract 1 from the available sub-channel number of $A_d^R(t)$ after $C_{t'}^R(l)$ is determined to be delivered.

Through the $\left|A_d^R\right|$ times iterations, we obtain modified T^R and modified A^R, N_{mul1} and V_{mul1}.

Algorithm 5 shows the Multicast Algorithm Based on Users and SBSs (MABUS).

3) MULTICAST ALGORITHM BASED ON MBS

The content items that have been cached in MBS can be delivered by multicasted or unicast. When multiple content items need be delivered from MBS, we first deliver the content items which are cached in local caching of MBS and are demanded by most requesting users by multicalst.

Input initial parameters. Define a row vector C_{K+1}^R, and find different content items $C_{d_0}^R$ in C_{K+1}^R. $C_{d_0}^R$ represents all demanded content items that are cached in local caching of MBS or remote servers. Define a row vector C_d^R, which represents all demanded content items that are cached in local caching of MBS. If $C_d^R \neq \emptyset$, we modify delivery scheme of demanded content items in C_d^R through following process.

Algorithm 5 Multicast Algorithm Based on Users and SBSs (MABUS)

Input: $\mathbf{V}, N^R, U^R, C^R, T^R, A^R, K, N, b_M, b_S$;
Output: $T^R, A^R, N_{mul1}, V_{mul1}$;

Define N^R dimensional row vectors N_{mul1}, V_{mul1} and $N+K+1$ dimensional row vectors S^{cha} as multicast cluster head vector, multicast threshold rate vector and available sub-channel number vector, then initial N_{mul1}, V_{mul1} as zero vector, and initialize $S^{cha} = [\text{ones}(1,K), b_M, b_S \times \text{ones}(1,N)]$;
Find $A_d^R = \text{unique}(A^R)$, which denotes all edge content delivery entities regardless of channel capacity;
for $t = 1 : \left|A_d^R\right|$
  Define two row vectors $U_t^R = \{r \,|A^R(r) = A_d^R(t)\,, \forall r = \{1, 2, \ldots, N^R\}\}$ and $C_t^R = \left\{i \,|i = C^R(r), A^R(r) = A_d^R(t),\ \forall r = \{1, 2, \ldots, N^R\}\right\}$, which denote requesting users and demanded content items corresponding to the t-th content delivery entities;
  Find $C_{t_0}^R = \text{unique}(C_t^R)$; sort content set $C_{t_0}^R$ in a descend order according to the number of different content items in C_t^R and obtain $C_{t'}^R$, then set $c_{t\max}^R = C_{t'}^R(1)$;
  Define two vectors $U_{t\max}^R = \{r_0 \,|r_0 \in U_t^R, C_t^R(r_0) = c_{t\max}^R\}$ and $U_{tnom}^R = U_t^R \backslash U_{t\max}^R$, where $U_{t\max}^R$ denotes requesting users in U_t^R that demand content item $c_{t\max}^R$;
  if $A_d^R(t) < K+1$
    if $\left|C_t^R\right| = 1$, then perform Case A;
    else Perform Case B;
    end if
  elseif $A_d^R(t) > K+1$
    for $l = 1 : \left|C_{t'}^R\right|$
      In order to deliver the l-th demanded content items cached in SBS $A_d^R(t)$, we define a requesting user vector $U_{tl}^R = \{r \,|r \in U_t^R, C^R(r) = C_{t'}^R(l)\}$;
      if $S^{cha}(A_d^R(t)) = 0$, then perform Case C;
      else
        if $\left|U_{tl}^R\right| = 1$, then perform Case D;
        else Perform Case E;
        end if
      end if
    end for
  end if
end for
Case A: if $\left|U_t^R\right| > 1$, the content item C_t^R is multicasted to requesting users of U_t^R (otherwise, C_t^R is unicasted to U_t^R). Set cluster head and max rate threshold corresponding to requesting users of U_t^R according to $A_d^R(t)$ and $\mathbf{V}$, i.e., set $N_{mul1}(U_t^R(n_1)) = A_d^R(t)$ and $V_{mul1}(U_t^R(n_1)) = \min_{k \in \left\{U^R(U_t^R(e_1)) \,|e_1 = 1,2,\ldots|U_t^R|\right\}} \{\mathbf{V}(k, A_d^R(t))\}$, for all $n_1 \in \{1, 2, \ldots, \left|U_t^R\right|\}$;

24: Case B: if $\left|U^R_{t\max}\right| > 1$, content item $c^R_{t\max}$ is delivered to requesting users of $U^R_{t\max}$ by multicast(otherwise by unicast). Similar to Case A, set cluster head and max rate threshold corresponding to requesting users of $U^R_{t\max}$.Due to the constraint of bandwidth capacity, requesting users of U^R_{tnom} fetch content items from remote servers via MBS, set $T^R(U^R_{tnom}(m_1)) = 2$ and $A^R(U^R_{tnom}(m_1)) = K+1$,for all $m_1 \in \{1, 2, \ldots, \left|U^R_{tnom}\right|\}$;

25: Case C: since available sub-channel number of $A^R_d(t)$ is 0, requesting users of U^R_{tl} fetch content items from remote servers. Set T^R and A^R of requesting users in U^R_{tl} similar to Case B;

26: Case D: when $T^R(U^R_{tl}) = 1$, $C^R_{t'}(l)$ is delivered directly from $A^R_d(t)$ to U^R_{tl} and the available sub-channel number of $A^R_d(t)$ needs to be reduced by 1 (i.e., $S^{cha}(A^R_d(t)) = S^{cha}(A^R_d(t))-1$);

27: Case E: content item $C^R_{t'}(l)$ is multicasted to requesting users of U^R_{tl}. Similar to Case A, set cluster head and max rate threshold corresponding to requesting users of U^R_{tl} and the available sub-channel number of $A^R_d(t)$ needs to be reduced by 1;

Define a row vector N^R_d and initialize it, whose m-th element $N^R_d(m)$ represents the number of the requesting users that fetch $C^R_d(m)$ from MBS. Afterward, we compute N^R_d and decide on N_{mul1} and V_{mul1} corresponding to all demanded content items in C^R_d through $\left|C^R_d\right|$ times iterations. For the m-th time iteration ($m = 1, 2, \ldots, \left|C^R_d\right|$), we define a row vector U^R_m which denotes all requesting users that fetch $C^R_d(m)$ from MBS. If $\left|U^R_m\right| > 1$, $C^R_d(m)$ can be delivered from MBS to requesting users in U^R_m by multicast, we set N_{mul1} and V_{mul1}, and then set $N^R_d(m) = \left|U^R_m\right|$. Sort N^R_d in a descend order and obtain vector $N^R_{d'}$. If $\left|C^R_d\right| > b_M$, deliver the content items in C^R_d whose corresponding $N^R_{d'}$ values are less than $N^R_{d'}(b_M)$ from remote server and deliver other in C^R_d direct from MBS. Through the above iterations, we obtain multicast delivery scheme at MBS and output T^R, N_{mul1} and V_{mul1}.

Algorithm 6 shows the Multicast Algorithm Based on MBS (MABM).

4) EDGE PRIORITY ALGORITHM BASED ON MULTICAST

Input initial parameters, and initialize $\mathbf{Z}$ and $\mathbf{Z}_B$. Afterward, we use EPARBC to obtain content delivery scheme regardless of the constraint of bandwidth capacity. Use MABUS and MABM to modify content delivery scheme obtained from EPARBC under the constraint of bandwidth capacity, and obtain U^R, T^R, A^R, N_{mul1} and V_{mul1}. At last, we use U^R, T^R and A^R to obtain $\mathbf{Z}$ and $\mathbf{Z}_B$ through the N^R times iterations.

Algorithm 7 shows the edge priority algorithm based on multicast (EPABM).

In the EPABM algorithm for the CDBM problem, there is at most N^R times iteration. Therefore, in the worst case, the computational complexity of the EPABM algorithm is $O(N)$.

Algorithm 6 Multicast Algorithm Based on MBS (MABM)

Input: $K, b_M, \mathbf{V}, N^R, D_B, L, U^R, C^R, T^R, A^R, N_{mul1}, V_{mul1}$;
Output: T^R, N_{mul1}, V_{mul1};

1: Define three vectors $C^R_{K+1} = \{i \,|\, i = C^R(r), A^R(r) = K+1, \forall r \in \{1, 2, \ldots, N^R\}\}$, $C^R_{d_0} = \text{unique}(C^R_{K+1})$ and $C^R_d = \{i \,|\, i = C^R_{d_0}(r), \mathbf{X}_{opt}(i, K+1) = 1, \forall r \in \{1, 2, \ldots, N^R\}\}$, where C^R_d represents all demanded content items that are cached in local caching of MBS;
2: **if** $C^R_d \neq \emptyset$
3: Define a $\left|C^R_d\right|$ dimensional vector N^R_d whose m-th element ($m = 1, 2, \ldots, \left|C^R_d\right|$) $N^R_d(m)$ represents the number of requ- esting users that fetch $C^R_d(m)$ from MBS, initialize N^R_d;
4: **for** $m = 1 : \left|C^R_d\right|$
5: Define a row vector $U^R_m = \{r \,|\, C^R(r) = C^R_d(m), T^R(r) = 1, A^R(r) = K+1, \forall r \in \{1, 2, \ldots, N^R\}\}$, which denotes requesting users that fetch $C^R_d(m)$ from MBS;
6: **if** $\left|U^R_m\right| > 1$
7: Content item $C^R_d(m)$ is delivered from MBS to requesting users of U^R_m by multicast, and set $N_{mul1}(U^R_m(n_2)) = K+1$, $V_{mul1}(U^R_m(n_2)) = \min_{k \in \{U^R(U^R_m(e_2)) \,|\, e_2 = 1,2,\ldots,\left|U^R_m\right|\}} \{\mathbf{V}(k, K+1)\}$ for all $n_2 \in \{1, 2, \ldots, \left|U^R_m\right|\}$;
8: **end if**
9: Set $N^R_d(m) = \left|U^R_m\right|$;
10: **end for**;
11: Sort N^R_d in a descend order and obtain vector $N^R_{d'}$;
12: **if** $\left|C^R_d\right| > b_M$
13: Then deliver content items in C^R_d whose corresponding $N^R_{d'}$ values are less than $N^R_{d'}(b_M)$ from remote servers, i.e.,set T^R values corresponding to these demanded content items as 2;
14: **end if**
15: **end if**

V. SIMULATION RESULTS

In this section, we evaluate the performance of our proposed extensive cooperative content caching and delivery scheme based on multicast. In order to demonstrate the effectiveness of our proposed scheme, we introduce two metrics include the average downloading delay and local caching hit rate. We define the average downloading delay as the average value of delay which all users have experienced in order to download their demanded content items, and define local caching hit rate as the ratio of the number of content requests which are served by local caching at D2D user equipments, SBSs and MBS to the total number of all requesting users' content requests[9].

We use the following schemes as comparison references to evaluate the caching scheme' performance of this paper. Specific comparison schemes are as follows:

Algorithm 7 Edge Priority Algorithm Based on Multicast (EPABM)

Input: $K, N, F, b_M, b_S, \mathbf{V}, \tau_{ki}, N^R, D_B, L, \mathbf{M}^{con}, \mathbf{X}_{opt}$;
Output: $\mathbf{Z}, \mathbf{Z}_B, N_{mul1}, V_{mul1}$;
1: Initialize $\mathbf{Z}$ and $\mathbf{Z}_B$;
2: Use EPARBC to obtain U^R, C^R, T^R and A^R regardless of the limit of bandwidth capacity;
3: Use MABUS to modify T^R, A^R, and obtain N_{mul1} and V_{mul1} according to multicast based on users and SBSs;
4: Use MABM to modify T^R, N_{mul1} and V_{mul1} according to multicast based on MBS;
5: **for** $r = 1 : N^R$
6: **case:** $T^R(r) = 1$, then set $\mathbf{Z}(U^R(r), A^R(r)) = 1$;
7: **case:** $T^R(r) = 2$, then set $\mathbf{Z}_B(U^R(r)) = 1$;
8: **end for**

- MBS no caching (MNC): MBS has no caching ability; SBSs and D2D users that a requesting user can be connected with can cooperatively cache content items for the requesting user according to the cooperative content caching (OC^3) scheme in [9]. When user u_k demands content item c_i, u_k chooses to fetch c_i from other users via D2D links or SBSs that u_k can be connected with; otherwise u_k must fetch c_i from the remote server via MBS. The above content delivery is formulated as an optimal content delivery scheme that is solved by Hungarian algorithm [9].
- MBS random caching and no cooperation (MRCNC): MBS has caching ability (but caching capacity is limited) in which content items are cached randomly; SBSs and D2D users that a requesting user can be connected with can cooperatively cache content items according to the OC^3 scheme in [9]. When a user u_k demands an arbitrary content item c_i, u_k first chooses to fetch c_i from other D2D users, SBSs or MBS that u_k can be connected with; otherwise u_k must fetch c_i from the remote server via MBS. The above content delivery is formulated as an optimal content delivery scheme that is solved by Hungarian algorithm [9].
- MBS popular caching and no cooperation (MPCNC): the content caching scheme and content delivery scheme of MPCNC are the same as MRCNC except that content items are cached in MBS according to popularity of content items [12]–[16], in which most popular content items are cached firstly in MBS.
- Greedy algorithm for D2D caching (GADC): D2D users that requesting user u_k can be connected with cooperatively cache content items for u_k, and an arbitrary content item c_i can only be cached in at most one of D2D user equipments of u_k. MBS and SBSs have no caching ability, but u_k can fetch all content items from remote servers via MBS. We use Greedy algorithm in [35] to solve the caching scheme. When u_k demands c_i, u_k first chooses to fetch c_i from the nearest D2D user that has cached c_i. When c_i has not been cached in all D2D users of u_k, u_k must fetch c_i from the remote server via MBS.

We consider a macrocell D2D-enabled HetNet which consists of a MBS, $N = 16$ SBSs and $K = 200$ users. The total number of content items is $F = 400$. The coverage radiuses of the MBS and each SBS are $r_M = 350$m and $r_S = 100$m [23], respectively. We locate MBS at the center of the whole area, and randomly distribute SBSs and users in the cell. We assume that the maximum distance of D2D communication is $r_U = 50$m [26]. The MBS, SBSs and users' transmission power is $P_U = 0.25$W, $P_S = 2$W and $P_M = 40$W [23], respectively. We assume the total bandwidth in the cell is 20MHz and the sub-channel bandwidth is $B_E = 180$KHz [23]. Since spatial sub-channel reuse between SBSs as assumed above, the number of sub-channels of each SBS and MBS can reach $b_S = 5$ and $b_M = 40$, respectively. The size of each content item is $L = 2048$bit, and the caching capacity of each user equipment, SBS and MBS is $C_U = 5$, $C_S = 30$ and $C_M = 60$ (unit is content item), respectively. We assume users' content request arrivals follow independently Poisson processes with user u_k's average request arrival rate $\lambda_k = 0.6$ (arrival /time period) for $\forall u_k \in \mathscr{U}$. The channel parameters are respectively path loss exponent $\alpha = 4$, power spectral density N_0 = - 174dbm/Hz, backhaul delay $D_B = 48$ms, minimum downloading rate $R_{\min} = 1.024 \times 10^5 bit/s$. The parameters in SA are respectively repeated cooling times $N^S = 30$, cooling coefficient $\alpha^S = 0.95$, maximum temperature $T_{\max} = 97$, minimum temperature $T_{\min} = 3$. The parameters in HGA are respectively population size $N^G_{gro} = 100$, evolution generation size $N^G_{gen} = 50$, number of selected chromosome $N^G_{sel} = 50$, mutation probability $p^G_{mut1} = 0.1$, Gene mutation probability $p^G_{mut2} = 0.01$. Since each caching entity has the same sub-channel bandwidth B_E, we substitute the number of sub-channel of SBSs for its bandwidth capacity to evaluate the performance of our proposed scheme. All simulation parameters are given in Table 2.

In Fig. 4, we compare the average downloading delay and local caching hit rate with the increase of the number of content items. From Fig. 4 we observe that the average downloading delay increases and the local caching hit rate decreases when the number of content items increases from 250 to 450, and this is due to the constraint of caching capacity in caching entities. We can also find that our proposed EC^3 scheme is more efficient than other four schemes in Fig. 4. We from Fig. 4 can see that the more content items are, the larger the performance gap between EC^3 scheme and other four schemes is. So our proposed EC^3 scheme is more efficient for more content items.

Fig. 5 shows the average downloading delay and local caching hit rate when the number of SBSs varies. From Figs. 5(a) and 5(b) we can find that the average downloading delay decreases and local caching hit rate increases with the number of SBSs ranging from 10 to 30 in MNC, MRCNC, MPCNC and EC^3 schemes, which is because there are more users that can be served by SBSs. The performance of our

TABLE 2. Simulation parameters.

Parameters	Values
Number of users	$K = 200$
Number of SBSs	$N = 16$
Number of content items	$F = 400$
The coverage radius of MBS	$r_M = 350$m
The coverage radius of each SBS	$r_S = 100$m
Maximum distance of D2D communication	$r_U = 50$m
The transmission power of MBS	$P_M = 40$W
The transmission power of each SBS	$P_S = 2$W
The transmission power of each user	$P_U = 0.25$W
The sub-channel bandwidth	$B_E = 180$KHz
Number of sub-channels of each SBS	$b_S = 5$
Number of sub-channels of MBS	$b_M = 40$
The size of each content item	$L = 2048$bit
The caching capacity of each user	$C_U = 5$ content item
The caching capacity of each SBS	$C_S = 30$ content item
The caching capacity of MBS	$C_M = 60$ content item
User u_k's average arrival rate	$\lambda_k = 0.6$
Path loss exponent	$\alpha = 4$
The power spectral density	N_0 = - 174dbm/Hz
Repeated cooling times in SA	$N^S = 30$
Cooling coefficient in SA	$\alpha^S = 0.95$
The maximum temperature in SA	$T_{\max} = 97$
The minimum temperature in SA	$T_{\min} = 3$
The population size in GA	$N^G_{gro} = 100$
The evolution generation size in GA	$N^G_{gen} = 50$
Number of selected chromosome in GA	$N^G_{sel} = 50$
The mutation probability in GA	$p^G_{mut1} = 0.1$
Gene mutation probability in GA	$p^G_{mut2} = 0.01$
The backhaul delay	$D_B = 48$ms
The minimum downloading rate	$R_{\min} = 1.024 \times 10^5 bit/s$

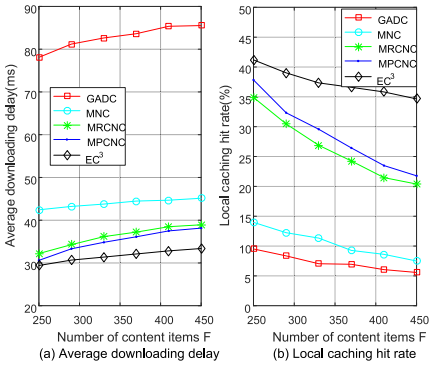


FIGURE 4. Average downloading delay and local caching hit rate versus the number of content items respectively.

proposed EC^3 scheme is much better than that of other four schemes on the average downloading delay and local caching hit rate. This is due to extensive cooperative caching in all caching entities, the content delivery based on multicast and the introduction of more accurate user CRP. In Fig. 5, we can find that the more SBSs are, the larger the performance gap between EC^3 scheme and other four schemes is. Therefore, our proposed EC^3 scheme is more efficient for more SBSs.

Figs. 6(a) and 6(b) show the average downloading delay and local caching hit rate versus the number of users respectively ($F = 1000$). As is shown in Fig. 6, we can find that the average downloading delay of EC^3 scheme decreases with the number of users ranging from 200 to 900; we also observe

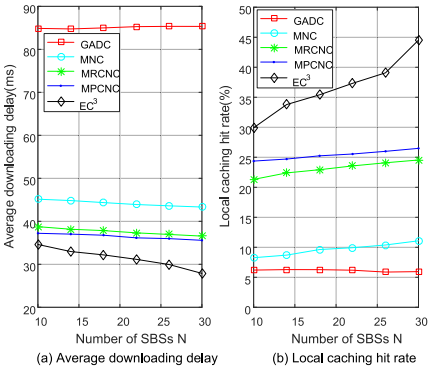


FIGURE 5. Average downloading delay and local caching hit rate versus the number of SBSs respectively.

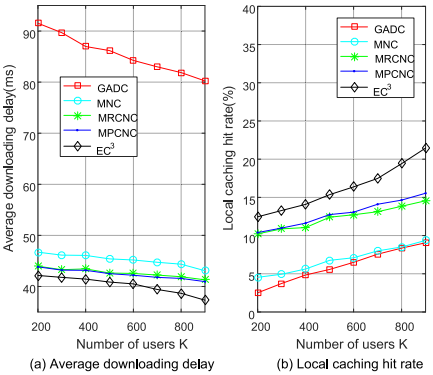


FIGURE 6. Average downloading delay and local caching hit rate versus the number of users respectively.

that the local caching hit rate of EC^3 scheme increases with the number of users ranging from 200 to 900. This is because users' intensity increases with the users' number and the increase of users' intensity leads to the more adjacent users that can be connected with and serve requesting users. In fact, under the condition of limited spectrum resources, the number of D2D users in MBS coverage cannot grow unlimited. When the density of D2D users reaches a certain limit, the limited spectrum capacity makes new D2D users hard be assigned link bandwidth through reusing; continuous increase in the number of users will make normal D2D communication hard be guaranteed due to the mutual interference between D2D users. Therefore, when the average download delay decreases to a certain value, the average download delay cannot go on decreasing with the increase of users' number. From Fig. 6, we can see that the change of performance on the average downloading delay and local caching hit rate in other four schemes is similar to that of EC^3 scheme. Besides, the performance of our proposed EC^3 scheme is much better than that of other four schemes on the average downloading delay and local caching hit rate.This is due to extensive cooperative caching in edge caching entities, and the introduction of a CRP for each user predicted by context information. From Fig. 6, it can be seen that to cache content items in MBS can decrease average downloading delay

and increase local caching hit rate by comparing MRCNC, MPCNC and EC^3 schemes with MNC scheme. Compared with MRCNC and MPCNC schemes, the average downloading delay of EC^3 scheme is lower and local caching hit rate of EC^3 scheme is higher, which shows that the performance can be further improved when we extensively consider the cooperative content caching among D2D user level, SBs level and MBS level. Compared with D2D user level cooperative caching in GADC scheme and two levels cooperative caching in MNC scheme, three levels cooperative caching in EC^3 scheme can most effectively reduce average downloading delay and improve local caching hit rate. In Fig. 6(a), we can find that the performance gap is larger for more users, and this is mainly because the number of the requesting users that participate in multicast increases with the increase of the number of users.

Figs. 7 and 8 show the average downloading delay and local caching hit rate versus the caching capacity of user equipments and SBSs respectively. Fig. 7 shows the average downloading delay decreases and the local caching hit rate increases when the caching capacity of D2D user equipments ranges from 5 to 25 content items in all five schemes, which is because more content items that the requesting users demand can be cached in D2D user equipments. The performance of our proposed EC^3 scheme is the best of that of all five schemes on the average downloading delay and local caching hit rate. Similar to Fig. 7, Fig. 8 shows the average downloading delay decreases and the local caching hit rate increases when the caching capacity of SBSs increases from 10 to 30 content items in MNC, MRCNC, MPCNC and EC^3 schemes, which is because there are more demanded content items that can be cached in SBSs. The performance of EC^3 scheme is still much better than that of other schemes.

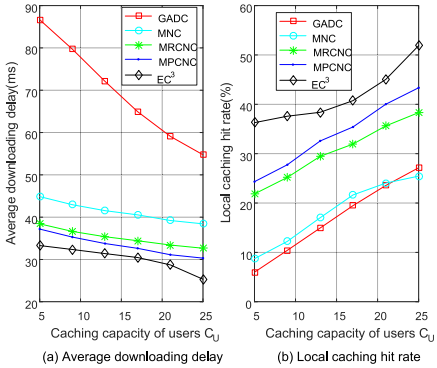


FIGURE 7. Average downloading delay and local caching hit rate versus the caching capacity of user equipments respectively.

Fig. 9 compares the average downloading delay and local caching hit rate when the caching capacity of MBS varies. Figs. 9(a) and 9(b) show the average downloading delay decreases and the local caching hit rate increases when the caching capacity of MBS ranges from 20 to 70 content items in MRCNC, MPCNC and EC^3 schemes, which is because MBS can cache more content items that requesting users

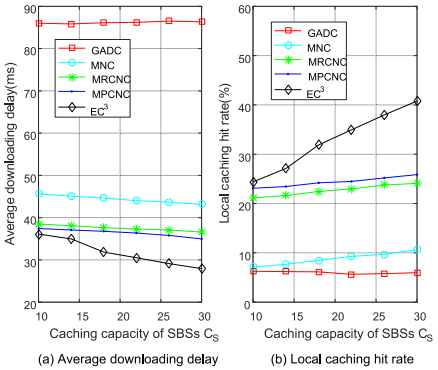


FIGURE 8. Average downloading delay and local caching hit rate versus the caching capacity of SBSs respectively.

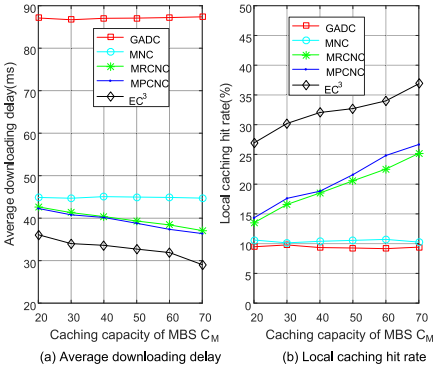


FIGURE 9. Average downloading delay and local caching hit rate versus the caching capacity of MBS respectively.

demand when the caching capacity of MBS increases. Similar to Figs. 7 and 8, the performance of EC^3 scheme is much better than that of other four schemes due to the extensive cooperative caching in all edge caching entities, the content delivery based on multicast and the introduction of more accurate user CRP.

Figs. 10(a) and 10(b) show the average downloading delay and local caching hit rate versus the number of sub-channels of SBSs respectively. From Figs. 10(a) and 10(b) we can find that the average downloading delay decreases and local

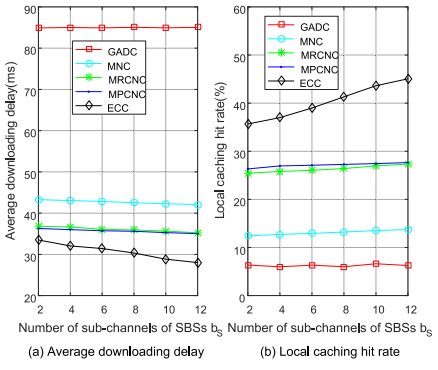


FIGURE 10. Average downloading delay and local caching hit rate versus the number of sub-channels of SBSs respectively.

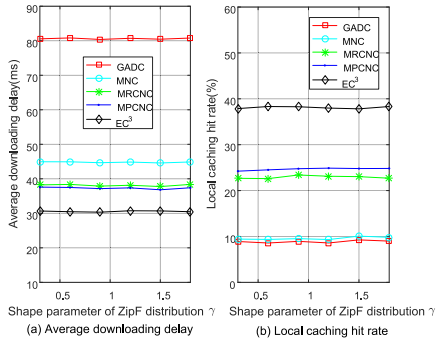


FIGURE 11. Average downloading delay and local caching hit rate versus different shape parameters of ZipF distribution respectively.

caching hit rate increases with the number of sub-channels of SBSs ranging from 2 to 12 in MNC, MRCNC, MPCNC and EC^3 schemes, which is because the requesting users have more sub-channels to fetch content items from SBSs. The performance of our proposed EC^3 scheme is much better than that of other four schemes. This is mainly due to the extensive cooperative caching in all edge caching entities and the introduction of more accurate user CRP.

Figs. 11(a) and 11(b) show the average downloading delay and local caching hit rate versus different shape parameters of ZipF distribution γ respectively. From Figs. 11, we can find that the average downloading delay and local caching hit rate are not nearly changed with shape parameter of ZipF distribution γ ranging from 0.3 to 1.8 in all five caching schemes, which is because the popularity of content items cannot reflect the preference of user when many requesting users' interests and preferences differ greatly. The preference of the users in a local region (e.g., in the coverage of a MBS) for content items may be greatly different, i.e., a large number of user requests cannot be concentrated on a small number of content items. In this case, ZipF distribution cannot accurately predict CRP of each user for all content items. Therefore, the increase of shape parameter of ZipF distribution γ cannot change the performance of the caching scheme. We can also see the performance of EC^3 scheme is much better than that of other four schemes due to the introduction of more accurate user CRP.

VI. CONCLUSION

In this paper, we propose an extensive cooperative content caching and delivery scheme based on multicast to address the contradiction between explosive growth of mobile data traffic and the demand for low delay and high user rate of experience in 5G HetNets. We firstly propose EC^3 scheme for D2D-enabled HetNets, which is solved by HGA. Furthermore, we develop CDBM scheme, which is solved by a suboptimal EPABM. Simulation results indicate that our proposed extensive cooperative content caching and delivery scheme based on multicast can significantly improve the system performance on the average downloading delay and caching hit rate compared with the existing cooperative content caching and delivery schemes. We can consider adding some caching capacity in MBS as a copy to back up some content items cached in D2D user equipments according to the average CRP of users in the coverage of MBS, which can alleviate the influence caused by user mobility and further reduce the average download delay. Though user equipments, SBSs and MBS have only a fixed caching capacity in our proposed EC^3 scheme, they can be directly extended to the case of variable caching capacity. Besides, adaptive resource allocation and privacy security in extensive cooperative content caching and delivery scheme are our future work.

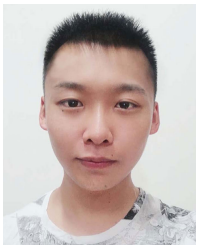
REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," Whiter Paper, 2017.
[2] J. Ma, J. Wang, and P. Fan, "A cooperation-based caching scheme for heterogeneous networks," *IEEE Access*, vol. 5, pp. 15013–15020, Dec. 2017.
[3] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.
[4] D. Wu, L. Zhou, and P. Lu, "Win-win driven D2D content sharing," *IEEE Internet Things J.*, early access, Nov. 30, 2020, doi: 10.1109/JIOT.2020.3041082.
[5] Q. C. Li, H. Niu, A. T. Papathanassiou, and G. Wu, "5G network capacity: Key elements and technologies," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 71–78, Mar. 2014.
[6] T. Zhang, X. Fang, Y. Liu, G. Y. Li, and W. Xu, "D2D-enabled mobile user edge caching: A multi-winner auction approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12314–12328, Dec. 2019.
[7] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for device-to-device caching networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 226–237, Mar. 2019.
[8] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.
[9] 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.
[10] 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.
[11] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. 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.
[12] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2894–2905, May 2014.
[13] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.* Oct. 2007, pp. 1–14.
[14] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[15] Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, "Probabilistic small-cell caching: Performance analysis and optimization," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4341–4354, May 2017.
[16] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1897–1903.
[17] X. Zhao, P. Yuan, H. Li, and S. Tang, "Collaborative edge caching in context-aware device-to-device networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9583–9596, Oct. 2018.
[18] Y. Wang, X. Tao, X. Zhang, and Y. Gu, "Cooperative caching placement in cache-enabled D2D underlaid cellular network," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1151–1154, May 2017.

[19] Y. Sun, Z. Chen, and H. Liu, "Delay analysis and optimization in cache-enabled multi-cell cooperative networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–7.
[20] 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.
[21] Y. Xu, X. Li, and J. Zhang, "Device-to-device content delivery in cellular networks: Multicast or unicast," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4401–4414, May 2018.
[22] Y. Wang, X. Tao, X. Zhang, and G. Mao, "Joint caching placement and user association for minimizing user download delay," *IEEE Access*, vol. 4, pp. 8625–8633, Dec. 2016.
[23] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
[24] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.
[25] R. Chai, Y. Li, and Q. Chen, "Joint cache partitioning, content placement, and user association for D2D-enabled heterogeneous cellular networks," *IEEE Access*, vol. 7, pp. 56642–56655, Feb. 2019.
[26] D. Feng, L. Lu, Y. Yuan-Wu, G. Li, S. Li, and G. Feng, "Device-to-device communications in cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 49–55, Apr. 2014.
[27] S. Singh, H. S. Dhillon, and J. G. Andrews, "Offloading in heterogeneous networks: Modeling, analysis, and design insights," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2484–2497, May 2013.
[28] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: Modeling, design and experimental results," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.
[29] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Comput. Oper. Res.*, vol. 35, no. 12, pp. 3860–3884, Dec. 2008.
[30] L. A. N. Lorena and M. G. Narciso, "Relaxation heuristics for a generalized assignment problem," *Eur. J. Oper. Res.*, vol. 91, no. 3, pp. 600–610, Jun. 1996.
[31] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics*, vol. 52, no. 1, pp. 7–21, Feb. 2005.
[32] S. Kirkpatrick, D. G. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated anneal," *Science*, vol. 42, no. 3, pp. 671–680, Jan. 1983.
[33] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, pp. 95–99, Oct. 1988.
[34] Y. R. Elhaddad, "Combined simulated annealing and genetic algorithm to solve optimization problems," World Acad. Sci., Eng. Technol., Paris, France, Tech. Rep., 2012, vol. 6, no. 8, pp. 1508–1510.
[35] J. Qu, D. Wu, Y. Long, W. Yang, and Y. Cai, "D2D based caching content placement in wireless cache-enabled networks," *J. Internet Technol.*, vol. 20, no. 2, pp. 333–344, 2019.

QINXUE FU received the B.S. degree from the Xi'an Telecommunications College of PLA, Xi'an, China, in 2003, and the M.S. degree from Xidian University, Xi'an, in 2006. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA, Nanjing, China. His current research interests include social-aware D2D communications, D2D resource management, and game theory.

LIANXIN YANG received the B.S. and M.S. degrees from the PLA University of Science and Technology, Nanjing, China, in 2015 and 2017, respectively. She is currently the Ph.D. degree with the Army Engineering University of PLA, Nanjing. Her current research interests include cross-modal communications, resource allocation, and machine learning.



BAOQUAN YU received the B.S. degree in communication engineering from the Institute of Communications Engineering, PLA Army Engineering University, Nanjing, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include short packet communications, machine type communications, and age of information.

YAN WU received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA, Nanjing. His current research interests include D2D communications, content sharing, game theory, and haptic communications.