Received 26 March 2020; revised 27 April 2020; accepted 29 April 2020. Date of publication 6 May 2020; date of current version 27 May 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.2992305

Traffic Offloading for Content Distribution Assisted With Device-to-Device Communications

WEI SONG (Senior Member, IEEE), AND HAORU XING

Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada

CORRESPONDING AUTHOR: W. SONG (e-mail: wsong@unb.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

ABSTRACT While mobile networks are evolving rapidly, the battle between ever-growing traffic demands and out-paced network capacities will continue and require more efficient solutions. Emerging techniques such as mobile edge computing and device-to-device (D2D) communications can help relieve traffic at the mobile edge and accommodate surging traffic demands from various content-centric services. In this work, we focus on exploiting device caching and user collaboration to offload content distribution traffic. Specifically, we investigate the request offloading problem, which aims to appropriately select caching devices and maximize the content requests that can be fulfilled through D2D communications. Given the constraints of individual transmission and caching capacities, the number of available D2D channels, and information privacy with social-awareness, we can decouple the request offloading problem into two subproblems, i.e., the device caching and matching problem, and the D2D channel allocation problem. As we prove that both problems are NP-hard, we propose efficient algorithms that iteratively make a best local decision in each step. Simulation results show that the proposed algorithms perform fairly closely to optimal solutions in small-scale instances and outperform the reference schemes under various situations.

INDEX TERMS Mobile edge computing, D2D communications, collaborative content distribution, device caching, device matching, and channel allocation.

I. INTRODUCTION

NOWADAYS our daily lives have been flooded with diverse mobile applications, ranging from entertainment and education to business and health. The proliferation of social media further boosts spreading of popular contents to mobile users and causes substantial duplicate transmissions. According to [1], mobile data traffic is expected to double every year. Particularly, video traffic has been dominant and accounted for 75% of global mobile data traffic in 2017. Mobile network providers need to deploy efficient technologies such as device-to-device (D2D) communications to maintain sustainable growth while improving user experience. D2D communications offer various benefits such as improving spectrum efficiency, offloading traffic or tasks, and enabling proximity services [2]–[5].

From the network perspective, there have been many studies in the literature that focus on D2D resource allocation in a general D2D network context [4]–[7]. A critical challenge to radio resource allocation with underlay spectrum sharing is how to control co-channel interference when D2D and cellular users are allowed access the same spectrum so as to achieve high spectrum efficiency. From the application perspective, D2D communications have been explored to support various applications such as content distribution, social networking and public safety services. Particularly, D2D communications can assist content distribution to offload traffic from the cellular network. For example, content requests can be offloaded from a base station (BS) by caching received contents in user devices and fulfilling other nearby requests via D2D forwarding. Device caching strategies are investigated in [8]–[10], assuming that D2D users are cooperative and willing to cache a video that is not requested by themselves. Social-awareness is further taken into account in [11], [12], so that D2D users are more incentivized to cache and forward contents for their social connections.

In this work, we intend to study D2D-assisted content distribution by integrating various considerations from both the network perspective and the application perspective. On one hand, D2D channel allocation should be extended from

the general context and tailored to meet specific application demands. On the other hand, content requests need to be effectively offloaded to D2D communications, and fulfilled by device caching and forwarding. Particularly, we attempt to address two key problems involved in request offloading for D2D-assisted content distribution. First, content placement (i.e., cache device selection) and content forwarding (i.e., device matching) can be considered together. Once we obtain a device matching solution that maximizes the fulfilled requests through D2D content sharing, we can naturally make decisions on cache device selection. Similarly, a source device can only fulfill no more than a certain number of requests from other devices due to the stint of device energy and bandwidth. Furthermore, because the maximum number of D2D users allowed in a cell is often limited to restrict the interference to regular cellular users, a subsequent problem is to properly allocate channels to matched D2D groups so that the intra-interference between D2D groups and inter-interference with cellular users are acceptable. Specifically, our main contributions are as follows:

- We investigate request offloading in D2D-assisted content distribution by decomposing it into two subproblems, i.e., the device caching and matching problem and the D2D channel allocation problem. Each subproblem is formulated as an integer linear program (ILP). We analyze their computational hardness and prove that both are NP-hard in the optimization form.
- To address the NP-hardness of the formulated problems, we propose effective algorithms to find approximate solutions. For the device caching and matching problem, the proposed algorithm consists of an iterative procedure for source selection and a post-processing step to augment device matching. We also obtain an upper-bound solution based on Lagrangian relaxation as a benchmark. For the D2D channel allocation problem, the proposed algorithm solves it as a generalized vertex coloring problem.
- We conduct simulations to evaluate and compare the performance of the proposed algorithms and the reference schemes. The results show that our algorithms achieve high performance close to that of the optimal solutions in small-scale cases and outperforms the reference schemes in larger-scale cases.

In the following, Section II reviews the related works on D2D content caching and forwarding. In Section III, we introduce the system model and the request offloading problem. Section IV formulates two subproblems for request offloading and analyzes their complexity. The proposed algorithms are discussed in Section V. Experimental results are given in Section VI. Section VII concludes this paper.

II. RELATED WORKS

In the literature, there have been many studies on the content caching strategies in D2D networks. For instance, an optimal content caching approach was proposed in [10] to maximize offloading probability with effective transmission accessibility between devices. In [13], the authors studied a probabilistic caching placement strategy for stochastic D2D caching networks, and obtained the optimal caching probabilities that maximize the requests successfully served by local device caches. Moreover, in [14], the authors optimized the strategies for cache device distribution for the purpose of maximizing the average density of successful receptions considering interference and noise.

In addition, there are studies on device pairing and content forwarding assisted with D2D communications. Different from the studies on content caching strategies, the focus here is the matching of request devices with devices that have already cached videos. For example, in [15], the authors exploited interference-aware collaborations among devices and proposed an approximate approach that aims to offload requests by pairing request devices with cache devices in close proximity. In [16], the approximate algorithm was further improved to a three-step approach for the device pairing problem to maximize the request coverage through D2D communications.

In [9], the authors studied edge caching at both BSs and user devices, where D2D communications can offload traffic to minimize transmission costs between BSs. A Q-learning based strategy was proposed for cache replacement at BSs. To take into account the effect of device caching and D2D offloading, they also formulated a maximum weighted independent sets problem and solved it by a greedy algorithm. Accordingly, they could obtain the probability that the requested contents are fulfilled by adjacent devices via D2D communications and thus relieved from the BS. In [17], a four-dimensional hypergraph model was used to address the problem of D2D pairing and channel allocation in cache-enabled D2D networks. The hypergraph model includes four dimensions with respect to content requesters, content helpers, resource blocks and cache contents. Aiming to maximize the sum of achievable data rate in the D2D network, a greedy algorithm was proposed to approach the optimal solution with low complexity.

Moreover, some studies take into account social-awareness in D2D content caching and sharing. These works intend to leverage users' attributes in both the physical network and the social network for content distribution. In [11], the authors proposed a D2D-assisted solution that profits from social relationships for source selection and data forwarding. In [18], a hypergraph model was proposed to incorporate multidimensional information including social-awareness. Then, hypergraph techniques such as coloring and matching can be used to optimize D2D spectrum management and cache placement. In [19], the authors applied such a hypergraph model for D2D channel allocation to coordinate the interference between D2D pairs and cellular users. A modified greedy hypergraph coloring algorithm is used to successively color the vertices corresponding to the D2D pairs and cellular users in a color (representing a channel) in a descending order of monodegree. As such, the cellular users and the D2D pairs are classified into clusters

with different colors, such that more users can be allocated to channels for a larger capacity.

Though social-awareness can potentially improve the performance of content distribution by caching and sharing, user devices are subject to various resource constraints, which should be considered in the caching and forwarding strategies. We notice that some research mainly considers the constraints from the network side, e.g., co-channel interference [15], [16], the service side, e.g., content requesting and caching patterns [13], [14], or the user side, e.g., social relationships and incentives [12], [18]. Nevertheless, this work intends to incorporate various constraints from different perspectives. Furthermore, many studies on D2D content sharing focus on one-to-one matching [9], [16], [17], [20]. We attempt to relax this constraint and consider one-to-many matching, which thereby accommodates more device heterogeneity in terms of resource capacity. Our problem is different from the regular one-to-many bipartite matching problem, which is solvable in polynomial time, in that one group of devices need to be split into a set of cache devices and a set of request devices instead of having two separate given sets of devices. Together with various other constraints, these extensions eventually cause the high complexity of our research problem, which will be shown with strict theoretical analysis. In view of the limited computation power of the mobile edge, we try to reduce the complexity by decomposition, and design fast and efficient algorithms to find reasonable solutions.

III. SYSTEM MODEL

A. CONTENT DISTRIBUTION SCENARIO WITH D2D-ASSISTANCE

In this work, we focus on a content distribution scenario at the mobile edge as depicted in Fig. 1. A set of request user devices, denoted by $\mathcal{N} = \{1, 2, \ldots, n\}$, are requesting contents from a library of m videos. All devices are uniformly located in the coverage of a BS, a circular area of radius L with the BS at the center. The distance between two devices i and j ($i, j \in \mathcal{N}$) is denoted by d_{ij}. Besides the physical attributes, we consider a social network among all users in $\mathcal{N}$, which is abstracted by graph $G_s = (\mathcal{N}, \mathcal{E}_s)$. If two users are socially connected, then an edge exists between the two corresponding nodes. Each device only requests one file from the video library and all requests form set $\mathcal{R}$ following a Zipf popularity distribution [9], [20]–[22]. Therefore, the probability that device $i \in \mathcal{N}$ requests video k is given by

$$p_i^k = \frac{\frac{1}{k^\sigma}}{\sum_{j=1}^{m} \frac{1}{j^\sigma}}, \quad \sigma > 0 \tag{1}$$

where exponent σ characterizes the relative popularity of videos. A higher value of σ indicates that user requests are concentrated on fewer videos. Let m_i denote the video requested by device i.

Video requests initiated by mobile devices are first submitted to the associated BS, which acts as an edge server for a number of devices within its coverage. The edge server can collect information of all request devices, provide solutions for video caching, and allocate D2D links for content sharing through the server's control functions. Considering that some popular videos are demanded by multiple neighbouring users, the edge server can screen the video requests and offload some to be fulfilled by cache devices over D2D links. This D2D-assisted content distribution not only can offload traffic from the mobile edge but also improves transmission efficiency with the close proximity.

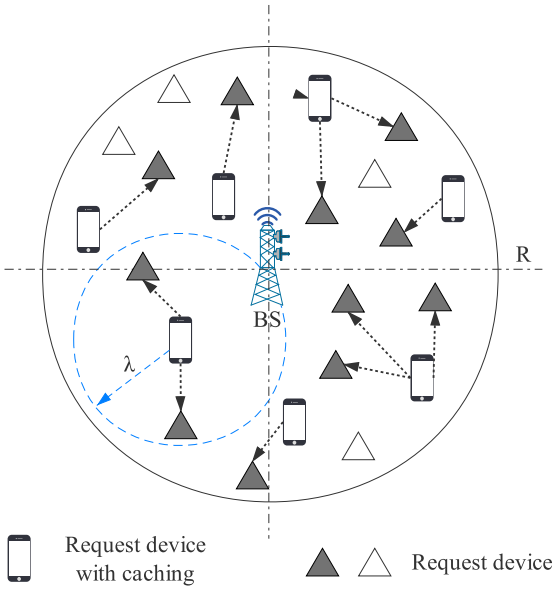

FIGURE 1. Content distribution scenario with request offloading.

B. REQUEST OFFLOADING TO D2D COMMUNICATIONS

Based on the content distribution scenario in Fig. 1, we study the request offloading problem, which aims to maximize the number of requests fulfilled by local cache devices (also called sources) via D2D links while satisfying certain resource and feasibility constraints.

- First, we limit the *collaboration distance* (denoted by λ) between devices to ensure the quality-of-service (QoS) of D2D communications. That is, a cache device can only serve request devices within its collaboration distance [23], [24]. In the literature, both D2D unicast and multicast have been studied for content distribution. As shown in [25], there are pros and cons for either method. The limited computation and communication capabilities of mobile devices pose intrinsic challenges to resource allocation and power control for D2D multicast. In view of the simplicity and feasibility of D2D unicast in current systems, we consider that a cache device unicasts the content to the request devices.
- In addition, each source $i \in \mathcal{N}$ can only fulfill no more than β_i requests from other devices due to the stint of battery life and network bandwidth. Moreover, as most users are unwilling to share contents with those they do not trust, we require that the users of selected

sources should be socially connected with their matched receivers to protect information privacy.
- Furthermore, the total number of selected sources is limited by γ. Although more caches lead to better offloading performance, it is infeasible to unrestrictedly allocate the edge resources for D2D transmissions. Each BS needs to reserve sufficient channel resources for other uses, such as for the regular cellular links. Therefore, we limit the total number of selected sources and thus restrict the number of occupied D2D channels.

Then, the *device caching and matching problem* aims to select at most γ sources from device set $\mathcal{N}$ to cache videos and match them to other request devices in $\mathcal{N}$. Rather than defining a one-on-one matching, we pursue a one-to-many matching solution that each transmitter can share its content to multiple receivers. Solving the device caching and matching problem, we can obtain at most γ D2D pairs or groups (with one D2D transmitter and one or multiple D2D receivers), which require at most γ distinct resource channels for the D2D links. Hence, we further address the *D2D channel allocation problem*. Based on the spatial distribution of the D2D users and underlaying cellular users that share their channels, we need to properly allocate the D2D channels to avoid co-channel interference and minimize channel occupation time so that these channels are released as early as possible for other uses.

It is seen in Fig. 1 that the system model considers a single-cell scenario. It is not hard to extend this work to a multi-cell scenario. We need to pay special attention to the devices near the cell edge. The solution can be adapted according to whether a cache device is allowed to serve request devices in multiple cells. Especially, inter-cell interference in addition to intra-cell interference should be taken into account.

IV. PROBLEM FORMULATIONS AND ANALYSIS

In this section, we formulate the two subproblems of request offloading, i.e., the device caching and matching problem and the D2D channel allocation problem and then analyze their computational complexity.

A. DEVICE CACHING AND MATCHING PROBLEM

Given the above content distribution scenario, we select both sources for content caching and their receivers from device set $\mathcal{N}$. Since every matching between sources and receivers must be feasible, we abstract the relationship between them with a bipartite graph $G_b = (\mathcal{N}, \mathcal{N}, \mathcal{L})$, which is illustrated in Fig. 2. Here, an edge indicates the feasibility of sharing video contents between two devices. An edge exists between devices i and j if they request the same video, fall within each other's collaboration distance, and are socially connected. Let binary variable l_{ij} represent whether an edge exists between devices i and j in the bipartite graph, i.e., $(i, j) \in \mathcal{L}$, defined as

$$l_{ij} = \begin{cases} 1, & \text{if } m_i = m_j, d_{ij} \le \lambda, \text{ and } e_{ij} \in \mathcal{E}_s \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

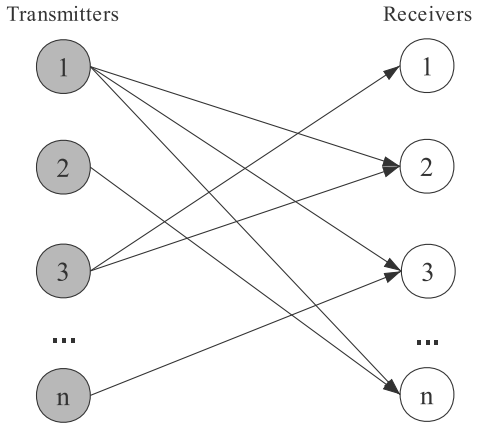

FIGURE 2. Bipartite graph model for transmitters and receivers.

Once an edge server receives all video requests from user devices, it filters requests to maximize the total number of requests that can be fulfilled by local device caching, while satisfying the constraints of D2D collaboration distance, budget of device caching, social relationship, and the number of simultaneous D2D links. Accordingly, we formulate the device caching and matching problem as the following ILP:

$$(P_1) \quad \max_{x,z} . \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} x_{ij} \tag{3a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} x_{ij} \le 1, \qquad \forall j \in \mathcal{N} \tag{3b}$$

$$\sum_{j\in\mathcal{N}} x_{ij} \le \beta_i, \qquad \forall i \in \mathcal{N} \tag{3c}$$

$$\sum_{i\in\mathcal{N}} z_i \le \gamma \tag{3d}$$

$$x_{ij} \le l_{ij}, \qquad \forall i, j \in \mathcal{N} \tag{3e}$$

$$\sum_{j\in\mathcal{N}} \left(-x_{ij}\right) \le M \cdot (1 - z_i) - 1, \quad \forall i \in \mathcal{N} \tag{3f}$$

$$\sum_{j\in\mathcal{N}} x_{ij} \le M \cdot z_i, \qquad \forall i \in \mathcal{N} \tag{3g}$$

$$\sum_{i\in\mathcal{N}} x_{ij} \le 1 - z_j, \qquad \forall j \in \mathcal{N} \tag{3h}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i, j \in \mathcal{N} \tag{3i}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in \mathcal{N}. \tag{3j}$$

Here, decision variable x_{ij} is a binary variable that indicates whether device i is selected to serve the request from device j. Hence, objective function (3a) maximizes the total number of requests to be fulfilled by selected sources. Constraint (3b) requires that each request should not be fulfilled by more than one source, constraint (3c) limits the serving budget of each device, and constraint (3d) indicates that the number of selected sources cannot exceed the maximum number of D2D links allowed in the cell. To ensure that the matching between sources and receivers is valid, constraint (3e) limits

the matching among the feasible edges in the bipartite graph in Fig. 2, which is equivalent to $x_{ij} = 0$, if $l_{ij} = 0, \forall i, j \in \mathcal{N}$.

In addition to x, problem (3) also involves binary decision variable z, where $z_i = 1$ indicates that device i is selected as a source to serve other request(s). Clearly, x and z are not independent but related by

$$z_i = \begin{cases} 0, & \text{if } \sum_{j\in\mathcal{N}} x_{ij} = 0 \\ 1, & \text{if } \sum_{j\in\mathcal{N}} x_{ij} \geq 1 \end{cases}, \quad \forall i \in \mathcal{N}. \tag{4}$$

In problem (3), conditional constraint (4) is expressed by inequality constraints (3f) and (3g), where M is a sufficiently large constant that meets $M \geq \max_i \beta_i$. Last, constraint (3h) narrows the relationship between x and z so that when a device is selected as a source it does not need to be further served by any other device. In Fig. 2, the set of nodes on the left is in fact the same set of nodes on the right. Hence, constraint (3h) defines that, if left node j is selected as a source, i.e., $z_j = 1$, the final matching should not include any edge incident on the corresponding right node j, i.e., $\sum_{i\in\mathcal{N}} x_{ij} = 0$.

B. D2D CHANNEL ALLOCATION PROBLEM

After solving the device caching and matching problem in (3), we can form at most γ D2D groups. According to the solution to problem (3), we can obtain the subset of selected sources (i.e., cache devices): $\mathcal{N}_t = \{i : z_i = 1, \forall i \in \mathcal{N}\}$. For each cache device $i \in \mathcal{N}_t$, the solution also gives the subset of request devices to serve: $\mathcal{N}_{r,i} = \{j : x_{ij} = 1, \forall j \in \mathcal{N}\}$. Each cache device and the corresponding matched request devices form a D2D group. Next, the BS needs to allocate channels for the D2D groups to forward the cached contents. This is the D2D channel allocation problem, which is to be formulated in this section in (5). To improve spectrum efficiency, these D2D users can reuse the channels allocated to certain underlaying cellular users if interferences can be avoided among them.

Fig. 3 illustrates an example for interferences among D2D groups. Assume that co-channel interference is unacceptable within transmission range λ but negligible beyond that. Consider a D2D group with source i and receiver set $\mathcal{N}_{r,i}$, and another D2D group with source j and receiver set $\mathcal{N}_{r,j}$. If at least one receiver device in $\mathcal{N}_{r,j}$ is located within the transmission range of source i, we say that these two D2D groups are in conflict. Similarly, such interference conflict may exist between a D2D group and a cellular user.

Then, depending on the spatial locations of D2D and cellular users, we obtain a conflict graph $G_c = (\mathcal{W} \cup \mathcal{N}_t, \mathcal{A})$ as shown in Fig. 4, where the vertices include the set of cellular users, $\mathcal{W}$, and the set of D2D groups, $\mathcal{N}_t$ (one vertex for each D2D group), and an edge exists between two vertices if they are interfering as shown in Fig. 3. In the conflict graph, each vertex needs to be labelled by a color representing its allocated channel. All cellular users have been labelled by distinct colors, which means they are allocated orthogonal channels. Then, each vertex corresponding to a D2D group needs to be assigned a color same as that

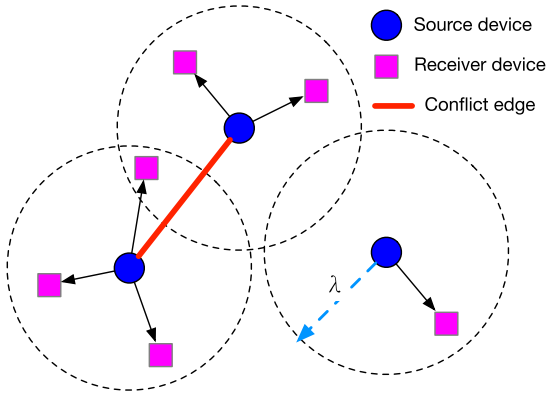

FIGURE 3. Illustration of interferences among D2D groups.

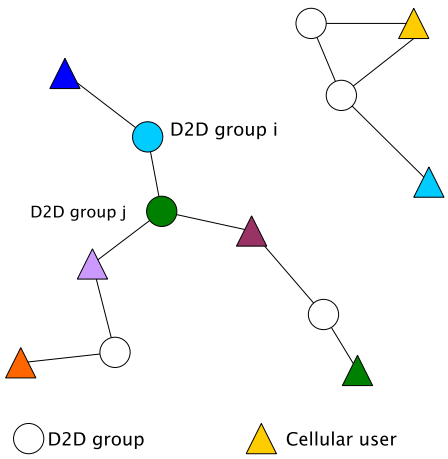

FIGURE 4. Conflict graph among D2D groups and cellular users.

of a cellular user, which means the D2D group reuses the cellular channel. However, any two vertices with a conflict edge cannot share the same channel (color label) to avoid interference in between.

Given that the data transmission for D2D group i lasts for duration τ_i, the D2D group will occupy the allocated channel for time τ_i. Here, transmission duration τ_i depends on the size of the requested video and the transmission rate over the D2D channel. The transmission rate is estimated according to the the maximum distance λ allowed between a cache device and the matched request devices. For one thing, we are mainly concerned with maximizing the number of offloaded requests instead of the sum rate of D2D communication devices. For another, it is challenging for a resource-limited user device to collect fine-grained channel conditions with the intended receivers. To minimize the total channel occupation time taken by all D2D groups, we formulate the D2D channel allocation problem as follows:

$$(P_2) \quad \min_{x,z} \; \sum_{k\in\mathcal{W}} z_k \tag{5a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{W}} x_{ki} = 1, \qquad \forall i \in \mathcal{N}_t \tag{5b}$$

$$x_{ki} + \zeta_{ij} x_{kj} \leq 1, \qquad \forall k \in \mathcal{W}, \quad \forall i, j \in \mathcal{N}_t \tag{5c}$$

$$x_{ki} \le 1 - \rho_{ki}, \qquad \forall k \in \mathcal{W}, \quad \forall i \in \mathcal{N}_t \tag{5d}$$
$$x_{ki}\tau_i \le z_k, \qquad \forall k \in \mathcal{W}, \quad \forall i \in \mathcal{N}_t \tag{5e}$$
$$x_{ki} \in \{0, 1\}, \qquad \forall k \in \mathcal{W}, \quad \forall i \in \mathcal{N}_t \tag{5f}$$
$$z_k \ge 0, \qquad \forall k \in \mathcal{W}. \tag{5g}$$

Here, decision variable x_{ki} indicates whether the channel of cellular user k is allocated to D2D group i. Hence, constraint (5b) requires that each D2D group be allocated one cellular channel. To avoid interference among D2D groups, constraint (5c) ensures that x_{ki} and x_{kj} for D2D groups i and j with a conflict edge (i.e., $\zeta_{ij} = 1$) cannot be set to one at the same time. To distinguish co-channel interference with cellular users, we use ρ_{ki} to represent the conflict between cellular user k and D2D group i. Then, constraint (5d) means x_{ki} must be zero when there exists interference conflict between cellular user k and D2D group i (i.e., $\rho_{ki} = 1$), which limits valid cellular channel candidates.

Last, (5e) defines another decision variable, z_k, which is the maximum transmission duration of all D2D groups that are allocated to reuse channel k. In other words, if channel k is allocated to multiple D2D groups, we should consider the longest time that channel k is occupied by any of these D2D groups, i.e., the makespan of the channel occupation time:

$$z_k = \max\{x_{k1}\tau_1, \ldots, x_{ki}\tau_i, \ldots, x_{k\gamma}\tau_\gamma\}. \tag{6}$$

Clearly, (6) can be represented by inequality constraint (5e). Therefore, objective function (5a) aims to minimize the total time span that the reused cellular channels are occupied. This design goal is different from that of many existing studies, which often focus on the total sum rates of D2D and cellular users [4], [5], [17]. Here, we consider makespan similar to virtualized resource allocation in cloud computing. That is, we view the cellular channels like shared computing power in the cloud. In cloud computing, service charge is usually by hours of usage. In our case, since the co-channel interference has been limited by avoiding the conflict edges to meet an acceptable QoS requirement, we are more concerned with how long the channel resources are occupied and thus cannot be used for other purposes. As seen, this is more aligned with the trend of network virtualization.

It is worth noting here that the D2D channel allocation problem in (5) is based on the worst-case estimate on the content transmission time. Thus, the problem can be decoupled from scheduling, which determines how the content is forwarded to multiple users sequentially. For one thing, this can reduce the problem complexity. For another, once a channel is allocated to a D2D group, existing D2D scheduling algorithms can be easily incorporated to further reduce the transmission time. In the literature, there have been many effective approaches on D2D scheduling such as [26], [27].

C. PROBLEM ANALYSIS AND HARDNESS RESULTS

In Sections IV-A and IV-B, we formulate two ILP problems for request offloading. In this section, we analyze their computational hardness and prove both are NP-hard.

Theorem 1: The device caching and matching problem in (3) is NP-hard [28].

In [28], we have proved that problem (3) is NP-hard, by reducing the well-known NP-hard maximum coverage problem (MCP) to a special instance of problem (3).

Theorem 2: The D2D channel allocation problem in (5) is NP-hard.

Proof: To prove problem (5) is NP-hard, we reduce the NP-hard vertex coloring problem (VCP) to a special case of (5). The VCP aims to use the minimum number of colors to mark a graph's vertices such that no two adjacent vertices are of the same color. Mathematically, the VCP can be formulated as

$$(VCP) \quad \min_{x,y} . \sum_{k \in \mathcal{W}} y_k \tag{7a}$$
$$\text{s.t.} \sum_{k \in \mathcal{W}} x_{ki} = 1, \qquad \forall i \in \mathcal{V} \tag{7b}$$
$$x_{ki} + \zeta_{ij} x_{kj} \le 1, \qquad \forall k \in \mathcal{W}, \quad \forall i, j \in \mathcal{V} \tag{7c}$$
$$x_{ki} \in \{0, 1\}, \qquad \forall k \in \mathcal{W}, \quad \forall i \in \mathcal{V} \tag{7d}$$
$$y_k \in \{0, 1\}, \qquad \forall k \in \mathcal{W}. \tag{7e}$$

Here, decision variable y_k indicates whether color $k \in \mathcal{W}$ is used, variable x_{ki} indicates whether color k marks vertex $i \in \mathcal{V}$, and parameter ζ_{ij} represents whether vertices i and j are adjacent in the graph.

Comparing problem (7) with the VCP in (5), we can map vertex set $\mathcal{V}$ in (7) to D2D group set $\mathcal{N}_t$ in (5), and decision variable y_k in (7) to z_k in (5). Then, it is easily seen that (5a) and (5b) are equivalent to (7a) and (7b), respectively. In addition, considering the special case with $\tau_i = 1, \forall i$, we have constraints (5c) and (5e) merged to one constraint in (7c). Moreover, assume that all cellular users are so far away from D2D groups that no interference occurs between cellular users and D2D users. In other words, all cellular channels in $\mathcal{W}$ are valid, so constraint (5d) is naturally satisfied.

As such, we construct an instance of D2D channel allocation problem (5) that is equivalent to the VCP in (7). Since the VCP is known to be NP-hard, problem (5) at least has the same complexity as the VCP and thus is also NP-hard. ■

V. PROPOSED SOLUTIONS

In this section, we propose efficient algorithms for the two NP-hard problems formulated in Section IV, and an upper-bound solution to problem (3) based on Lagrangian relaxation.

A. UPPER-BOUND SOLUTION TO DEVICE CACHING AND MATCHING

As given in Section IV-C, the device caching and matching problem is NP-hard. Although some small-scale instances can be solved according to the ILP formulation by ILP solvers such as [29], it is computationally infeasible when a large number of user devices are associated with each

edge server. Here, for comparison purpose, we first derive an upper bound by Lagrangian relaxation, and use it as a benchmark when an optimal solution is intractable.

Relaxing the difficult constraints (3f), (3g), and (3h) of problem (3), we obtain the *Lagrangian dual* as follows:

$$
\begin{aligned}
(D)\quad \min_{\mu,\nu,\eta}.\ \ & C^*(\mu,\nu,\eta) \triangleq \max_{x,z}. \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} x_{ij} \\
& + \sum_{i\in\mathcal{N}} \mu_i \left(\sum_{j\in\mathcal{N}} x_{ij} - 1 + M\cdot(1-z_i) \right) \\
& + \sum_{i\in\mathcal{N}} \nu_i \left(M\cdot z_i - \sum_{j\in\mathcal{N}} x_{ij} \right) \\
& + \sum_{j\in\mathcal{N}} \eta_j \left(1 - z_j - \sum_{i\in\mathcal{N}} x_{ij} \right) \\
\text{s.t.}\quad & \sum_{i\in\mathcal{N}} x_{ij} \le 1, \qquad \forall j \in \mathcal{N} \\
& \sum_{j\in\mathcal{N}} x_{ij} \le \beta_i, \qquad \forall i \in \mathcal{N} \\
& \sum_{i\in\mathcal{N}} z_i \le \gamma \\
& x_{ij} \le l_{ij}, \qquad \forall i,j \in \mathcal{N} \\
& x_{ij} \in \{0,1\}, \qquad \forall i,j \in \mathcal{N} \qquad (8)
\end{aligned}
$$

where the *Lagrange multipliers* are $\mu = \{\mu_1, \ldots, \mu_{|\mathcal{N}|}\} \ge 0$, $\nu = \{\nu_1, \ldots, \nu_{|\mathcal{N}|}\} \ge 0$, and $\eta = \{\eta_1, \ldots, \eta_{|\mathcal{N}|}\} \ge 0$.

The *Lagrangian subproblem* $C^*(\mu,\nu,\eta)$ can be decomposed into two independent subproblems:

$$
\begin{aligned}
(S_1)\quad \max_{x}.\ \ & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} x_{ij}\big(1+\mu_i-\nu_i-\eta_j\big) \qquad (9) \\
\text{s.t.}\quad & \sum_{i\in\mathcal{N}} x_{ij} \le 1, \qquad \forall j \in \mathcal{N} \\
& \sum_{j\in\mathcal{N}} x_{ij} \le \beta_i, \qquad \forall i \in \mathcal{N} \\
& x_{ij} \le l_{ij}, \qquad \forall i,j \in \mathcal{N} \\
& x_{ij} \in \{0,1\}, \qquad \forall i,j \in \mathcal{N} \\
(S_2)\quad \max_{z}.\ \ & \sum_{i\in\mathcal{N}} z_i(M\cdot\nu_i - M\cdot\mu_i - \eta_i) \qquad (10) \\
\text{s.t.}\quad & \sum_{i\in\mathcal{N}} z_i \le \gamma \\
& z_i \in \{0,1\}, \qquad \forall i \in \mathcal{N}.
\end{aligned}
$$

Here, subproblem S_1 can be viewed as an instance of the well-known transportation problem [30]. Specifically, the unit gain over each edge of the bipartite graph in Fig. 2 is defined by weight $(1+\mu_i-\nu_i-\eta_j)$, the demand of each left node i is limited by β_i, and the capacity of each right node is just one. As an instance of the transportation problem, S_1 can be solved in polynomial time by ILP solvers. On the other hand, subproblem S_2 can be easily solved by ranking $|\mathcal{N}|$ weights, $(M\cdot\nu_i - M\cdot\mu_i - \eta_i)$, in a descending order and setting z_i to one only if the corresponding weight is positive, until there are no more positive weights or γ of the $|\mathcal{N}|$ variables of z_i have been set to one. After solving S_1 and S_2, we have the optimal solution, denoted by x^* and z^*, to Lagrangian subproblem $C^*(\mu,\nu,\eta)$ with given Lagrange multipliers μ, ν, and η. The optimal value is given by

$$
\begin{aligned}
C^*(\mu,\nu,\eta) = & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} x^*_{ij}\big(1+\mu_i-\nu_i-\eta_j\big) \\
& + \sum_{i\in\mathcal{N}} z^*_i(M\cdot\nu_i - M\cdot\mu_i - \eta_i) \\
& + \sum_{i\in\mathcal{N}} ((M-1)\cdot\mu_i + \eta_i). \qquad (11)
\end{aligned}
$$

Next, in order to solve the Lagrangian dual in (8), we need to search for the best Lagrange multipliers that minimize the objective value $C^*(\mu,\nu,\eta)$. As such, we can obtain the tightest upper bound to the optimal value of primal problem P_1 before Lagrangian relaxation. Algorithm 1 shows the details of solving the Lagrangian dual with the subgradient method. Here, the values of all Lagrangian multipliers are first initialized. Then, Lagrangian subproblem $C^*(\mu^{(t)},\nu^{(t)},\eta^{(t)})$ is solved with current Lagrangian multipliers in each iteration t. Accordingly, the Lagrangian multipliers are updated to

$$
\begin{aligned}
\mu_i^{(t+1)} &= \max\left\{0, \mu_i^{(t)} - \alpha^{(t)}\left(\sum_{j\in\mathcal{N}} x^*_{ij} - 1 + M\cdot\big(1-z^*_i\big)\right)\right\} \\
\nu_i^{(t+1)} &= \max\left\{0, \nu_i^{(t)} - \alpha^{(t)}\left(M\cdot z^*_i - \sum_{j\in\mathcal{N}} x^*_{ij}\right)\right\} \\
\eta_j^{(t+1)} &= \max\left\{0, \eta_j^{(t)} - \alpha^{(t)}\left(1 - z^*_j - \sum_{i\in\mathcal{N}} x^*_{ij}\right)\right\}
\end{aligned}
$$

where $\alpha^{(t)}$ is the step size for iteration t, given by $\alpha^{(t)} = \frac{1}{2^t}$. The iteration is terminated when the gap between $C^*(\mu^{(t)},\nu^{(t)},\eta^{(t)})$ and $C^*(\mu^{(t-1)},\nu^{(t-1)},\eta^{(t-1)})$ is not more than an accuracy threshold ϵ. The solution with the minimum objective value for the Lagrangian dual provides an upper bound for problem (3).

Here, we do not intend to use Algorithm 1 for practical application, but only consider it to obtain an upper bound for performance comparison. When the optimal solution is not available for large-scale instances, the upper-bound solution can be used as a benchmark. Generally, the subgradient method is guaranteed to converge even for non-differentiable objective function [31]. However, the convergence rate varies with the updating of step size $\alpha^{(t)}$ (Line 14) and the accuracy threshold ϵ (Line 11). To control the running time, we can limit the maximum number of iterations for the while-loop in Lines 5-15 by T. Inside the while-loop, all lines except Line 6 take a constant running time. Line 6 needs to obtain the optimal solutions to subproblems S_1

Algorithm 1: An Upper Bound for Device Caching and Matching With Lagrangian Relaxation

Input: Device set $\mathcal{N}$, social graph $G_s = (\mathcal{N}, \mathcal{E}_s)$, request set $\mathcal{R}$, $\{\beta_i : \forall i \in \mathcal{N}\}$, λ, γ, ϵ, T
Output: Solution $\tilde{x} = \{\tilde{x}_{ij} : \forall i, j \in \mathcal{N}\}$, objective $\tilde{\psi}$

1 Initialize $\{l_{ij} : \forall i, j \in \mathcal{N}\}$ with G_s, $\mathcal{R}$, and λ;
2 $C^* \leftarrow +\infty$;
3 $t \leftarrow 0, \alpha^{(t)} \leftarrow 1$;
4 $\mu^{(t)} \leftarrow \{0, \ldots, 0\}$, $\nu^{(t)} \leftarrow \{0, \ldots, 0\}$, $\eta^{(t)} \leftarrow \{0, \ldots, 0\}$;
5 **while** $t < T$ **do**
6 Obtain optimal solutions x^* and z^* to S_1 and S_2;
7 Compute objective value $C^*(\mu^{(t)}, \nu^{(t)}, \eta^{(t)})$;
8 **if** $C^*(\mu^{(t)}, \nu^{(t)}, \eta^{(t)}) < C^*$ **then**
9 $C^* \leftarrow C^*(\mu^{(t)}, \nu^{(t)}, \eta^{(t)})$;
10 $\tilde{x}_{ij} \leftarrow x^*_{ij}, \forall i, j \in \mathcal{N}$;
11 **if** $|C^*(\mu^{(t)}, \nu^{(t)}, \eta^{(t)}) - C^*(\mu^{(t-1)}, \nu^{(t-1)}, \eta^{(t-1)})| \leq \epsilon$ **then**
12 break;
13 Update Lagrange multipliers $\mu^{(t+1)}$, $\nu^{(t+1)}$, and $\eta^{(t+1)}$ according to the subgradient method;
14 Update step size: $\alpha^{(t+1)} \leftarrow \frac{1}{2^{(t+1)}}$;
15 $t \leftarrow t + 1$;
16 Compute objective value to primal problem P_1: $\tilde{\psi} \leftarrow \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \tilde{x}_{ij}$;
17 Return $\tilde{x}$, $\tilde{\psi}$;

and S_2. The time complexity for solving S_1 depends on the algorithm used to solve the underlying transportation problem. For example, if the algorithm in [30] is used, it takes time $O(|\mathcal{N}| \log(|\mathcal{N}|) \cdot (|\mathcal{L}| + |\mathcal{N}| \log(|\mathcal{N}|)))$ to solve problem S_1. Solving problem S_2 is essentially sorting $|\mathcal{N}|$ weights and selecting at most γ positive weights. The corresponding time complexity is $O(|\mathcal{N}| \log(|\mathcal{N}|))$. Therefore, Line 6 takes a total computing time of $O(|\mathcal{N}| \log(|\mathcal{N}|) \cdot (|\mathcal{L}| + |\mathcal{N}| \log(|\mathcal{N}|)))$. Thus, the overall time complexity of Algorithm 1 is $O(T \cdot |\mathcal{N}| \log(|\mathcal{N}|) \cdot (|\mathcal{L}| + |\mathcal{N}| \log(|\mathcal{N}|)))$.

B. PROPOSED ALGORITHM FOR DEVICE CACHING AND MATCHING

In Section V-A, since we relax the constraints that limit the relationships between x and z, the upper bound obtained by Lagrangian relaxation may not be feasible with respect to primal problem (3). Considering that (3) is NP-hard, we cannot obtain a feasible optimal solution in polynomial time. Hence, we propose Algorithm 2 to find an approximate solution.

The goal of problem (3) is to maximize the request coverage through source selection and device matching. In other words, we need to select at most γ sources among all devices and match them with other intended receivers so that we can maximize the total number of video requests that can be fulfilled by device caching. As seen in Algorithm 2, we rank all devices based on a heuristic metric h_i, which depends on the resource budget β_i of an individual device i, and the number of potential receivers (denoted by set S_i) that each device i can serve. Specifically, $S_i = \{j : l_{ij} = 1, \forall j \in \mathcal{N}\}$. Accordingly, we define metric h_i of device i as

$$h_i = \min\{\beta_i, |S_i|\}, \quad \forall i \in \mathcal{N}. \tag{13}$$

Here, by choosing the minimum value between β_i and $|S_i|$, we obtain the maximum number of requests that device i can serve if it is selected for video caching. Even if device i has a large value of $|S_i|$, it may not be able to share contents with other devices because of very limited transmission power (i.e., $\beta_i = 0$). Similarly, device i cannot fulfill any request if β_i is large but $S_i = \varnothing$. Therefore, we consider both the individual resource budget and the number of available candidates.

In Algorithm 2, we divide the solution into two main decision steps, i.e., caching source selection and device matching. First, in Lines 6-15, we select the sources according to their coverage capability (i.e., the heuristic metric). Since our goal is to select sources to fulfill as many requests as possible through device content sharing, we prefer to choose the devices that contribute more to request offloading. Hence, the device with the maximum heuristic metric is first selected as a source. In contrast, when choosing the receivers, we tend to prioritize the receivers with the fewest options, i.e., to match a source with the receiver that has the lowest capability of content caching and transmitting (Lines 10-13). In each round, a new source is selected while at most h_i receivers are tentatively matched to selected source i. It is worth noting that both sources and receivers are chosen from ranked candidate set $\mathbb{N}$, and a device should be removed from $\mathbb{N}$ if it has been selected as a source or a receiver. The heuristic metrics of the remaining devices in $\mathbb{N}$ are dynamically updated after each iteration.

Second, in Lines 16-21, the selected sources are matched to the potential receivers in an optimal manner. Though the procedure in Lines 6-15 tentatively assigns certain receivers to the selected sources, this matching may not be optimal due to the iterative procedure. Once the sources are determined, we can further obtain the optimal matching by formulating a transportation problem. Here, we modify the original bipartite graph in Fig. 2 by keeping only the edges incident on the selected sources on the left with $z_i = 1$, and also removing the edges incident on the right nodes corresponding to the selected sources. As such, we can ensure that the resulting matching can only be from a source device to a non-source receiver. Based on the modified bipartite graph, the transportation problem aims to maximize the number of covered requests (i.e., $\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} x_{ij}$) subject to the constraints of subproblem S_1 in (9). Then, the optimal matching with the sources can be obtained by solving the transportation problem optimally.

Next, we analyze the time complexity of Algorithm 2. Obviously, the intialization steps in Lines 1-5 take less computation time than the subsequent two main procedures in Lines 6-21. The first procedure for source selection

Algorithm 2: Device Caching and Matching Algorithm

Input: Device set $\mathcal{N}$, social graph $G_s = (\mathcal{N}, \mathcal{E}_s)$, request set $\mathcal{R}$, $\{\beta_i : \forall i \in \mathcal{N}\}$, λ, γ
Output: Solution $x* = \left\{x_{ij}^* : \forall i, j \in \mathcal{N}\right\}$, objective ψ^*

1 Initialize $\left\{l_{ij} : i, j \in \mathcal{N}\right\}$ with G_s, $\mathcal{R}$, and λ;
2 Initialize $x_{ij} \leftarrow 0, z_i \leftarrow 0, \forall i, j \in \mathcal{N}$;
3 Initialize potential receiver sets $\mathcal{S} = \{S_i : \forall i \in \mathcal{N}\}$;
4 Calculate heuristic metric h_i for each device $i \in \mathcal{N}$;
5 Rank devices in $\mathcal{N}$ in descending order of h_i and include those with $|S_i| \geq 1$ to set $\mathbb{N}$;
6 **begin** Select sources for local device caching
7 **while** $\sum_{i\in\mathcal{N}} z_i < \gamma$ *or* $\mathbb{N} \neq \varnothing$ **do**
8 Select top ranked device i in $\mathbb{N}$ as source: $z_i \leftarrow 1$;
9 Remove device i from $\mathbb{N}$;
10 **while** $\sum_{j\in\mathcal{N}} x_{ij} < \beta_i$ *or* $S_i \neq \varnothing$ **do**
11 Select device j with min. heuristic metric from potential receiver set S_i as receiver: $x_{ij} \leftarrow 1$;
 `// Device j cannot be a source any more`
12 Remove device j from $\mathbb{N}$;
13 Remove device j from all sets in $\mathcal{S}$;
14 Remove S_i from $\mathcal{S}$;
15 Update metric h_i for each device $i \in \mathbb{N}$ and re-rank $\mathbb{N}$ accordingly;
16 **begin** Optimize device matching with selected sources
 `// Modify bipartite graph based on selected sources`
17 Set $l_{i'j} \leftarrow 0, \forall i', i, j \in \mathcal{N}$, $i' \neq i$ and $z_i = 1$;
18 Set $l_{ij} \leftarrow 0, \forall i, j \in \mathcal{N}$, and $z_j = 1$;
19 Formulate a transportation problem with the modified bipartite graph, where the objective function is $\max . \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{ij}$ s.t. constraints in S_1;
20 Solve the transportation problem to find the optimal matching solution $\{x_{ij}^* : \forall i, j \in \mathcal{N}\}$;
21 Compute objective value to primal problem P_1: $\psi^* \leftarrow \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{ij}^*$;
22 Return x^*, ψ^*;

contains two nested loops and an inline re-ranking operation in Line 15. Thus, this procedure takes time $O(|\mathcal{N}| \cdot (|\mathcal{N}| + |\mathcal{N}| \log(|\mathcal{N}|)) = O(|\mathcal{N}|^2 \log(|\mathcal{N}|))$. The second procedure for optimizing device matching involves solving a transportation problem, which takes a polynomial time that depends on the specific algorithm. For example, the algorithm in [30] for the transportation problem runs in time $O(|\mathcal{N}| \log(|\mathcal{N}|) \cdot (|\mathcal{L}| + |\mathcal{N}| \log(|\mathcal{N}|)))$. Since the time complexity of the second procedure is dominant, it is also the overall worst-case time complexity of Algorithm 2.

Algorithm 3: D2D Channel Allocation Algorithm

Input: Set of D2D groups $\mathcal{N}_t$, set of cellular users $\mathcal{W}$, λ, transmission durations $\{\tau_i : \forall i \in \mathcal{N}_t\}$
Output: Solution $x^* = \left\{x_{ki}^* : \forall k \in \mathcal{W}, \forall i \in \mathcal{N}_t\right\}$, objective value ξ^*

1 Initialize conflict graph G_c among D2D groups and cellular users, configure $\left\{\zeta_{ij} : \forall i, j \in \mathcal{N}_t\right\}$ and $\{\rho_{ki} : \forall k \in \mathcal{W}, \forall i \in \mathcal{N}_t\}$;
2 Rank sources in $\mathcal{N}_t$ to set $\mathbb{N}_\mathrm{t}$ in descending order of $\{\tau_i : \forall i \in \mathcal{N}_t\}$ if $\tau_i \neq \tau_j$ and ascending order of vertex degrees in G_d with only D2D groups if $\tau_i = \tau_j$;
3 Rank cellular users in $\mathcal{W}$ to set $\mathbb{W}$ in ascending order of vertex degrees in G_c;
4 Initialize set of fulfilled sources $\mathcal{F} \leftarrow \varnothing$;
5 **begin** Allocate channels for sources
6 **while** $\mathbb{N}_\mathrm{t} \neq \varnothing$ **do**
7 Select top-ranked source i in $\mathbb{N}_\mathrm{t}$;
8 Assign top-ranked cellular channel $k \in \mathbb{W}$ without interference with i to i: $x_{ki}^* \leftarrow 1$;
9 Remove source i from $\mathbb{N}_\mathrm{t}$: $\mathbb{N}_\mathrm{t} \leftarrow \mathbb{N}_\mathrm{t} \setminus \{i\}$;
10 Add i to $\mathcal{F}$: $\mathcal{F} \leftarrow \mathcal{F} \cup \{i\}$;
11 Remove i from $\mathbb{N}_\mathrm{t}$: $\mathbb{N}_\mathrm{t} \leftarrow \mathbb{N}_\mathrm{t} \setminus \{i\}$;
12 for *all unfulfilled sources in* $\mathbb{N}_\mathrm{t}$ do
13 Select next ranked source j in $\mathbb{N}_\mathrm{t}$;
14 **if** *device j has no interference with cellular user k and all sources in* $\mathcal{F}$ **then**
15 Assign channel k to source j: $x_{kj}^* \leftarrow 1$;
16 Remove j from $\mathbb{N}_\mathrm{t}$: $\mathbb{N}_\mathrm{t} \leftarrow \mathbb{N}_\mathrm{t} \setminus \{j\}$;
17 Add j to $\mathcal{F}$: $\mathcal{F} \leftarrow \mathcal{F} \cup \{j\}$;
18 Calculate: $z_k^* \leftarrow \max\{x_{ki}^* \cdot \tau_i, \forall i \in \mathcal{F}\}$;
19 Remove cellular channel k from $\mathbb{W}$ and re-rank it: $\mathbb{W} \leftarrow \mathbb{W} \setminus \{k\}$;
20 Calculate objective value: $\xi^* \leftarrow \sum_{k\in\mathcal{W}} z_k^*$;
21 Return x^*, ξ^*;

C. PROPOSED ALGORITHM FOR D2D CHANNEL ALLOCATION

As proved in Section IV-C, the D2D channel allocation problem in (5) can be viewed as a generalized VCP and thus is NP-hard. Since there does not exist a polynomial-time algorithm to find the optimal solution, we propose Algorithm 3 to obtain a feasible and efficient solution. In problem (5), since we aim to minimize the total channel occupation time, a source with a long service duration can be prioritized to select a channel. Otherwise, if a source with a really long data transmission time is assigned a channel very late, it is likely that there would be fewer channel options and the total time span may be prolonged significantly. Furthermore, the channel of the cellular user who has the lowest vertex degree in the conflict graph is preferred because it has the minimal probability of interfering with sources. Therefore, at the beginning of Algorithm 3 in

Lines 2-3, we rank all sources in the descending order of their required data transmission time, and rank all cellular users in the ascending order of their vertex degrees in the conflict graph.

After that, the channel allocation procedure is divided into two nested loops in Lines 5-19. First, we consider the source that has not been assigned to any cellular channel and has the maximum required transmission time. Then, we iterate the ranked set of cellular users, find the first cellular user without interference with the source, and assign its channel to the source. This source is then removed from ranked set $\mathbb{N}_t$, but recorded in another temporary set $\mathcal{F}$, which maintains the sources that have been assigned cellular channels.

Since we have started to reuse a new cellular channel for a source device, it would be efficient to assign more remaining sources to share the same channel if possible. In Lines 12-17, we iterate the remaining sources in set $\mathbb{N}_t$. As long as an unfulfilled source has no interference with the currently selected cellular user and the sources already in fulfilled set $\mathcal{F}$, the unfulfilled source is also assigned to share this cellular channel. Meanwhile, the newly assigned source is added into $\mathcal{F}$ and removed from $\mathbb{N}_t$. As seen, through the two steps in each iteration, we attempt to make most use of a selected cellular channel and accommodate as many D2D receivers as possible with it. After that, this cellular channel is removed from the candidate set $\mathbb{W}$. This channel allocation procedure repeats until all sources are fulfilled and set $\mathbb{N}_t$ is cleared.

Next, we analyze the time complexity of Algorithm 3, which contains the initialization steps in Lines 1-4 and a subsequent channel allocation procedure in Lines 5-19. Noting that $|\mathcal{N}_t| \leq \gamma$, we have the running time of the initial steps as $O(\gamma \log \gamma + |\mathcal{W}| \cdot \log(|\mathcal{W}|))$. Since there are often much more cellular users than the allowed D2D users, i.e., $|\mathcal{W}| \gg \gamma$, the time complexity of the initial steps is $O(|\mathcal{W}| \cdot \log(|\mathcal{W}|))$. Next, we consider the two nested loops for the main channel allocation procedure. As the outer while-loop contains a ranking update in Line 19, this procedure has a time complexity $O(\gamma \cdot (|\mathcal{W}| \cdot \log(|\mathcal{W}|) + \gamma)) = O(\gamma \cdot |\mathcal{W}| \cdot \log(|\mathcal{W}|))$. Clearly, this procedure dominates the total running time of Algorithm 3, so its overall worst-case time complexity is $O(\gamma \cdot |\mathcal{W}| \cdot \log(|\mathcal{W}|))$.

VI. SIMULATION RESULTS AND DISCUSSIONS

This section evaluates the performance of the proposed algorithms under various problem scales and system settings.

A. SIMULATION SETTINGS

We develop a simulator in Java to evaluate the performance of the proposed algorithms for request offloading. Consider the content distribution scenario depicted in Fig. 1. A number of devices are randomly distributed within the circular area covered by a BS. The social relationships among the devices are simulated by the Erdos-Renyi model [32]. The videos requested by the devices follow the popularity model with a Zipf distribution [9], [20]–[22]. The content size is modelled by a heavy-tailed Weibull distribution, which has been suggested in previous analyses and measurements from YouTube videos [33], [34]. For simplicity, we fix the maximum D2D transmit power and the number of resource blocks for each D2D channel. Then, we estimate the D2D transmission rate by the Shannon limit considering the worst case with the maximum allowed collaboration distance λ. For each channel allocated to a D2D group, the channel occupation time depends on the video size, the D2D transmission rate, and the group size. As the same channel can be allocated to multiple conflict-free D2D groups to improve reuse efficiency, the overall occupation time of the channel is the maximum transmission duration of the corresponding D2D groups. In the following simulations, the BS or the edge server can conduct the device matching and channel allocation periodically. In each simulation round, the BS collects new information of devices and video requests, and then runs the algorithms for device matching and channel allocation accordingly. Table 1 gives the default parameters used in the following simulations.

TABLE 1. Simulation parameters.

Parameter	Value
Number of user devices	30, 200
Number of cache devices	15, 30
Max. number of served devices per cache	5
Shape parameter of Weibull distribution	0.5
Minimum size of videos	50 Mbits
Maximum size of videos	2 Gbits
Mean size of videos	1 Gbits
Coefficient of Zipf distribution	0.8
Probability for Erdo-Renyi model	0.9
D2D collaboration distance	110 m
Bandwidth of resource block	180 kHz
Resource blocks per D2D channel	6
D2D transmit power	20 dBm
Noise power density	-164 dBm/Hz
Path loss model (dB) with distance d in km	$128.1 + (37.6 \log_{10}(d))$

For Algorithm 2, we compare it with the optimal solution in small-scale cases. Although the device caching and matching problem is shown to be NP-hard, we can still obtain the optimal solution with some ILP solvers when the problem size is relatively small. When the optimal solution is intractable in large-scale cases, we consider the upper bound obtained by Lagrangian relaxation in Section V-A. As the upper-bound solution is not always feasible since certain constraints are relaxed, we mainly include it as a benchmark in the absence of the optimal solution. Similarly, Algorithm 3 is also compared with the optimal solution in small-scale cases.

In addition, we consider some reference schemes inspired by the related works such as [9], [17], [19], [20].

In [17], [20], some greedy schemes are used to pair D2D transmitters and receivers. Extending these ideas from one-to-one matching to one-to-many matching, we consider a reference scheme for our device caching and matching problem. It successively ranks user devices according to their numbers of potential receivers, and selects the device that covers the maximum number of uncovered requests as a source until γ devices are selected. For each newly selected source i, this scheme randomly assigns at most β_i receivers from the available set to fulfill their video requests. For the D2D channel allocation problem, we consider another reference scheme inspired by the idea in [9] based on the maximum independent set (MIS) problem. Referring to the conflict graph in Fig. 4, this scheme iteratively chooses an uncolored vertex according to an ascending order of the vertex degrees and constructs a maximal independent set that shares at least one available cellular channel. Once a new independent set is formed, a common cellular channel is assigned to this set and the scheme continues to find the next independent set. This procedure continues until all D2D vertices are colored or there is no more available channel.

B. PERFORMANCE OF DEVICE CACHING AND MATCHING SCHEMES

We first consider a small-scale scenario, where 30 user devices are randomly located around the BS in a cell area of radius 200m. The collaboration distance between two devices is set to 60m, and the maximum number of user requests that each device can serve is set between 0 and 5 (i.e., $\beta_i \in [0, 5]$, $\forall i$). The total number of selected sources is set to 10, which means that at most 10 devices can be selected to cache video contents. The social connections among the devices are simulated by following the Erdos-Renyi model [32] with connection probability 0.9. We run 50 rounds of experiments and the simulation scenario is randomly updated for each round.

Fig. 5(a) shows the total number of video requests that can be fulfilled by device caching with different solutions in this small-scale cases. As seen, the proposed Algorithm 2 performs better than the reference approach (labelled as "Matching extended" in the figures). On the other hand, compared with the optimal solution, the proposed algorithm achieves the optimal result in 39 rounds totally, while its objective values are only 1 or 2 fewer than those of the optimal solution in the other rounds. Based on the above results, we can see that our proposed solution performs fairly close to the optimal solution in the small-scale cases.

We further consider a larger-scale network, where there are 200 devices randomly located within a cell, and the collaboration distance is the same as above. We set the maximum number of sources to 30. As shown in Fig. 5(b), the gap between the proposed Algorithm 2 and the reference approach becomes much larger in each round. On average, 58.8% of video requests can be fulfilled through device caching under the proposed scheme, while the reference approach only offloads 36.4% of the video requests to

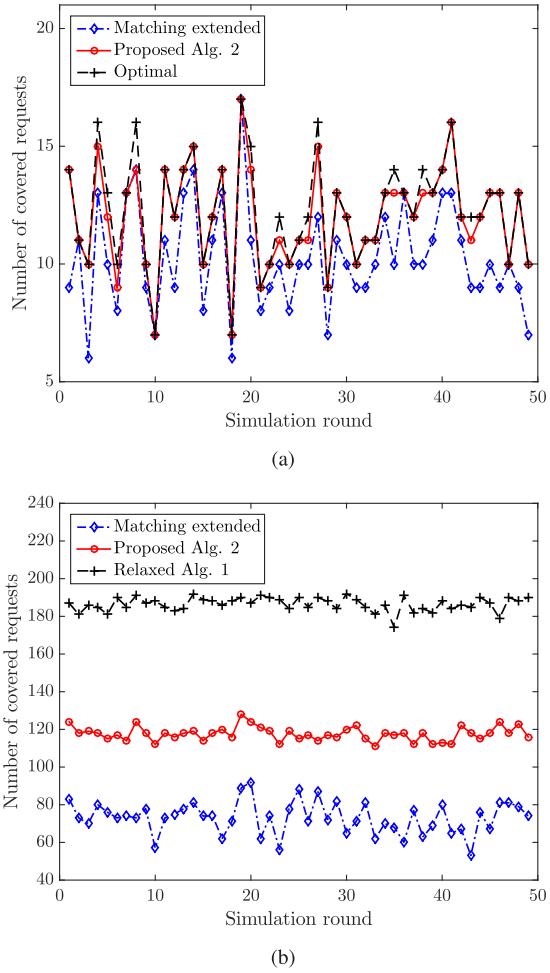

FIGURE 5. Performance of device caching and matching schemes. (a) In a small-scale network. (b) In a large-scale network.

D2D transmissions. Moreover, since the Lagrange relaxed solution provides an upper bound for the intractable optimal solution, we can see that the proposed scheme achieves at least 63% of the performance of the optimal solution.

Fig. 6 shows more statistics of the results with different device caching and matching schemes. As seen in Fig. 6(a), the proposed Algorithm 2 has a similar median as the optimal solution and spreads over a smaller range. In contrast, the result of the reference approach covers a range of lower bounds, which is more evident in Fig. 6(b) for the large-scale scenario.

Furthermore, we show the computing time of different device caching and matching schemes in Fig. 7. As seen, the reference approach takes the lowest average running time of less than one millisecond in the small-scale scenario and around 4 milliseconds in the large-scale scenario. The average computing time of the proposed Algorithm 2 is around 11 milliseconds in the small-scale scenario and in the order of hundreds of milliseconds in the large-scale scenario. Thus, we know that both schemes are feasible for practical application. On the other hand, the running time of the optimal solution and Algorithm 1 to obtain the upper-bound solution

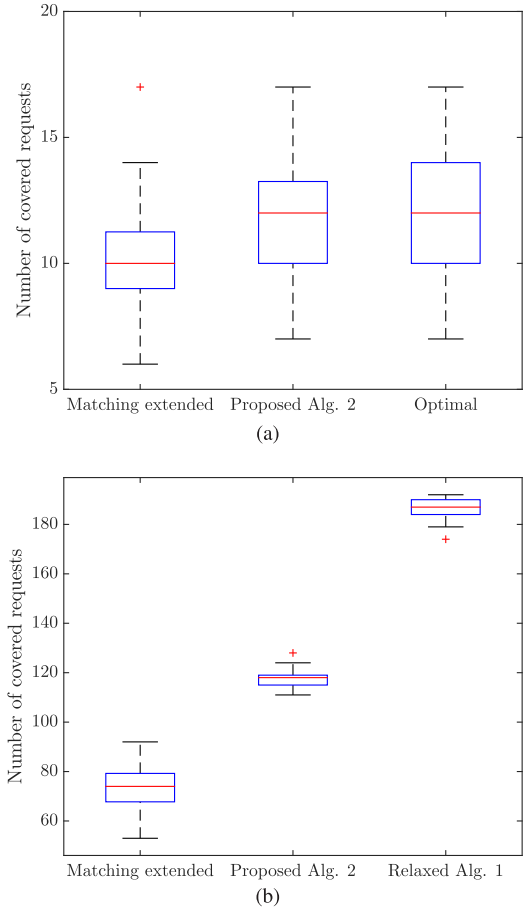

FIGURE 6. Result statistics of device caching and matching schemes. (a) In a small-scale network. (b) In a large-scale network.

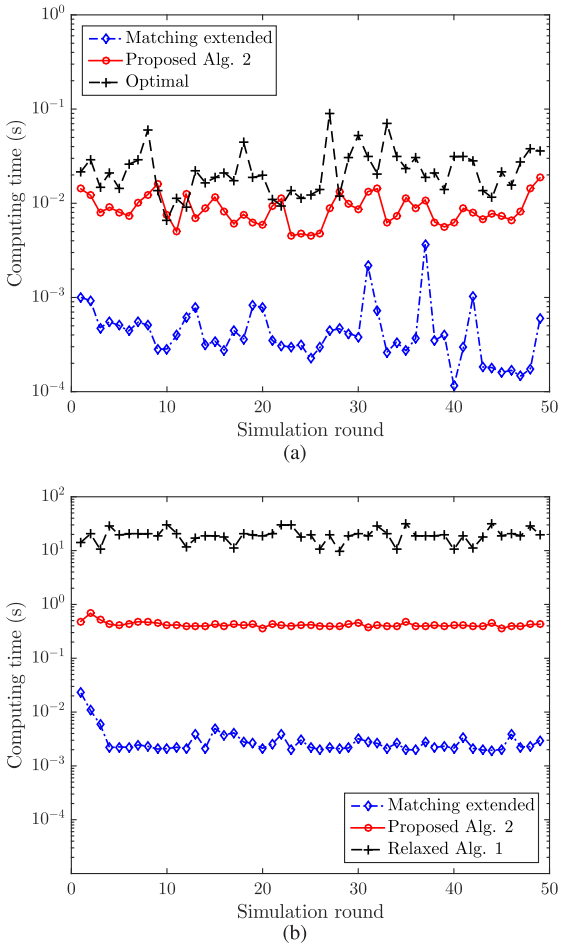

FIGURE 7. Computing time of device caching and matching schemes. (a) In a small-scale network. (b) In a large-scale network.

is much higher. On average, the optimal solution takes around 0.03 seconds for each small-scale instance in Fig. 7(a), while Algorithm 1 takes around 19.7 seconds to obtain an upper bound for each large-scale instance in Fig. 7(b). As mentioned in Section V-A, we only consider the upper-bound solution by Algorithm 1 as a benchmark for comparison, and do not intend to apply it to real systems.

C. PERFORMANCE OF D2D CHANNEL ALLOCATION SCHEMES

By device caching and matching, selective requests can be offloaded from the BS and fulfilled by D2D communications. To effectively reuse cellular channels, we propose Algorithm 3 for D2D channel allocation, which is compared with the optimal solution and the reference algorithm (labelled as "MIS based" in the figures). First, we consider a small-scale network with 30 user devices and the same settings as Fig. 5(a).

Fig. 8(a) shows the total channel occupation time with different channel allocation schemes. As seen, the proposed algorithm achieves the optimal values in all but one simulation round, which indicates the high efficiency of the proposed scheme in the small-scale network. In contrast, the reference scheme ends up with over 14.9% more total occupation time. Fig. 8(b) further shows the number of channels allocated for D2D transmissions in each round. We can see that the three schemes use a similar number of channels for D2D communications. However, as these schemes assign different channels to the D2D transmitters, their channel occupation time still varies even with the small-scale setting.

Fig. 9(a) further shows the total channel occupation time in the large-scale scenario with 200 devices as considered in Fig. 5(b). In comparison with the small-scale cases, more significant improvement is observed with the proposed algorithm over the reference scheme. Specifically, the proposed scheme occupies the allocated channels for approximately 31.4% less time than the reference scheme on average. Fig. 9(b) further shows that the proposed allocation scheme also reduces the cumulative channel occupation time for D2D transmissions, which significantly saves spectrum resources for other wireless services while meeting the content distribution requirements. This means that channel sharing for D2D content distribution can potentially achieve higher benefits when there is a larger user population.

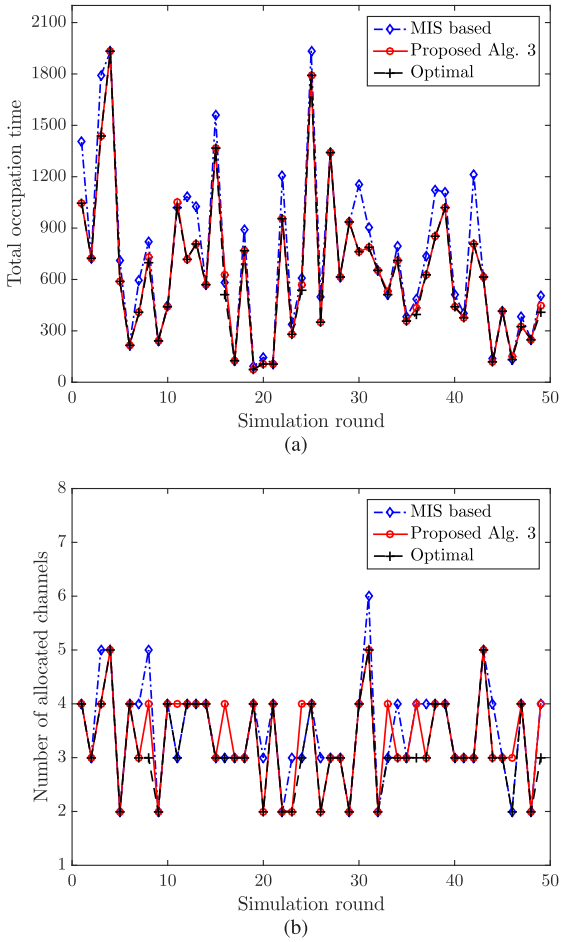

FIGURE 8. Performance of D2D channel allocation schemes in a small-scale network. (a) Total occupation time. (b) Number of allocated channels.

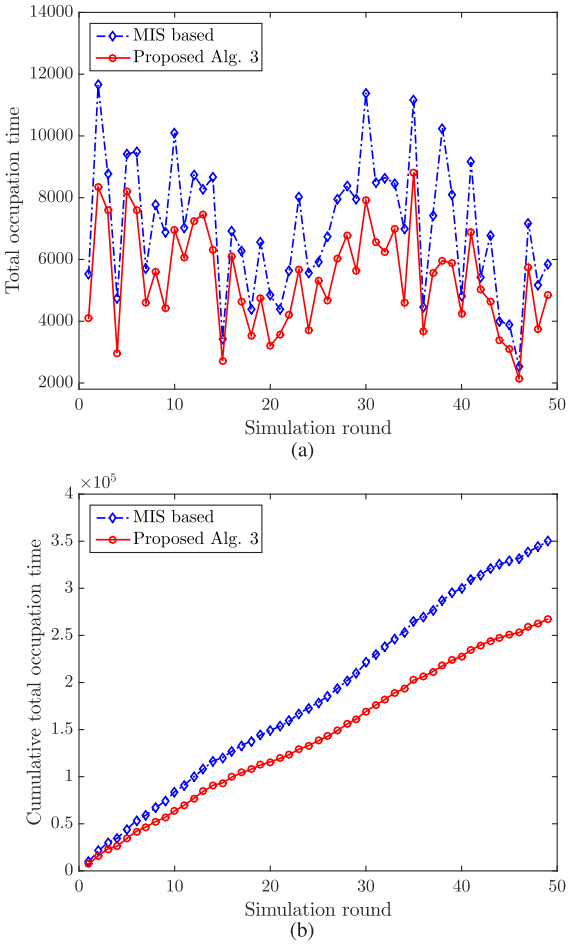

FIGURE 9. Performance of D2D channel allocation schemes in a large-scale network. (a) Total channel occupation time of each round. (b) Cumulative of total occupation time.

D. IMPACT OF COLLABORATION DISTANCE

For the simulation of this section, we assume all parameters except for collaboration distance λ follow the same setting as the large-scale scenario for Fig. 5(b). Fig. 10 illustrates how the performance varies with the collaboration distance. We can see that, with a larger collaboration distance, the gap between the proposed scheme and the reference scheme increases in general but with some minor fluctuations. This is because, when we have a larger collaboration distance, the edge server has more different options for cache device selection, i.e., as for which devices should be chosen to cache video contents and deliver them to other users. On the contrary, when there is a smaller collaboration distance, each candidate device has a lower capability of sharing contents with other devices, since they can only transmit videos to others located in the near vicinity. Hence, the results show little difference under short collaboration distances. In contrast, when it is affordable to share contents between distant users, the proposed algorithm can provide a more efficient strategy for device caching selection. Moreover, it is observed that the proposed scheme follows the trend of the relaxed upper-bound solution, whereas the reference approach shows more fluctuations. This is because the reference approach involves some randomness, while the proposed scheme is deterministic and thus more stable than the reference approach.

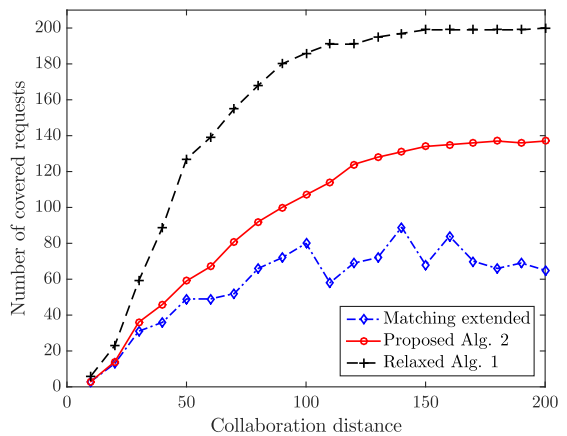

FIGURE 10. Performance variation with collaboration distance λ.

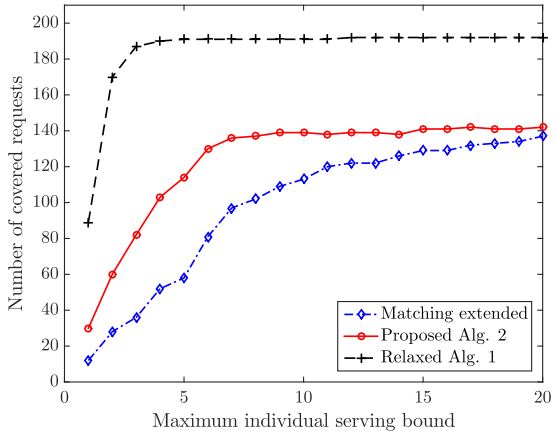

FIGURE 11. Performance variation with maximum budget $\max_i \beta_i$.

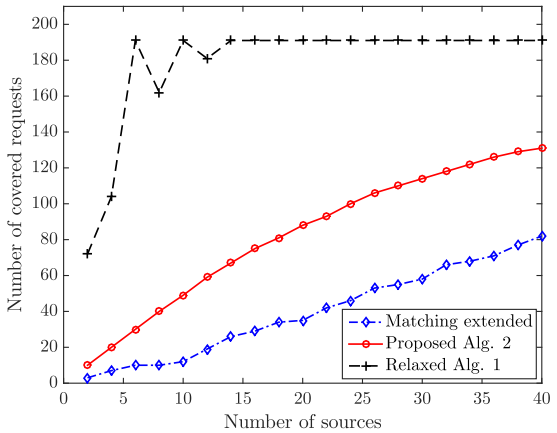

FIGURE 12. Performance variation with D2D capacity limit γ.

E. IMPACT OF RESOURCE BUDGET OF INDIVIDUAL DEVICE

In practice, uses devices are heterogeneous in terms of their capability of sharing video contents with others due to various concerns (e.g., battery life, private information security, and device caching capacity). Hence, we take into account the factor $\{\beta_i, \forall i\}$ in order to represent the resource constraint of each individual device for content caching and D2D forwarding. Assume that the resource budgets of all devices follow a uniform distribution within a range of $[0, b]$, where b is the maximum value of β_i. In this experiment, we vary b from 0 to 20, while keeping the other parameters the same as the large-scale scenario for Fig. 5(b). As shown in Fig. 11, the proposed algorithm covers more requests than the reference scheme in most cases. Especially when the devices do not have rich resources for sharing, our proposed solution can offload more traffic to the user plane. This is because it is able to identify stronger devices to cache and video contents with weaker ones. In contrast, when the device resources become abundant, the difference becomes smaller since it is easier to find an available source to cover a request.

Moreover, Fig. 11 shows that the number of covered requests with the proposed scheme increases significantly at the beginning, but the growth becomes flattened gradually after that. Similar to Fig. 10, we can observe that the performance growth of the proposed algorithm is smoother than that of the reference solution. On one hand, this is due to the randomness in the reference scheme. On the other hand, the available budget of each device is randomly generated and thus not increasing deterministically even if the maximum budget is larger. Consequently, the available resources are not guaranteed to satisfy more demands. Even with the minor fluctuations, we can still observe the general trend clearly.

F. IMPACT OF TRANSMITTER DENSITY

To show how transmitter density affects requests coverage, Fig. 12 shows how the performance varies with the maximum number of selected sources (i.e., γ). Here, we also consider the large-scale scenario for Fig. 11 but fix the maximum resource budget of each device to 5. Compared with the reference approach, the proposed scheme can cover over 41 more video requests on average while selecting the same number of transmitters.

VII. CONCLUSION AND FUTURE WORK

In this paper, we investigated a request offloading problem for collaborative content distribution, which takes advantage of device caching and D2D forwarding to achieve high efficiency. We addressed critical constraints from various perspectives while maximizing request offloading to D2D transmissions. To mitigate the complexity, we decomposed the request offloading problem into two subproblems. The first subproblem for device caching and matching selects certain devices as sources to temporarily store the received contents, and each source forms a group to fulfill the nearby members' content requests via D2D forwarding. Different from previous works, we jointly considered constraints from the network side, the service side, and the user side. The second subproblem allocates reused cellular channels for these D2D groups to provide their required resources while causing minimum interferences. Instead of focusing on the traditional design goals for D2D resource allocation, we targeted at the makespan of channel occupation and viewed the problem from the angle of virtualized resource sharing. As both subproblems are proved to be NP-hard, we developed effective algorithms to find approximate solutions. We conducted simulations with a wide range of system settings to examine their performance in different situations. The results show that the proposed algorithms can approach the optimal solutions in a small-scale network and significantly outperforms the reference schemes in a large-scale network. In this work, the proposed solution is network-controlled. In the future, it would be interesting to explore distributed solutions, e.g., by using federated learning [35], [36].

REFERENCES

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022*, Cisco, San Jose, CA, USA, 2019.
[2] W. Song and W. Zhuang, "Packet assignment under resource constraints with D2D communications," *IEEE Netw.*, vol. 30, no. 5, pp. 54–60, Sep./Oct. 2016.

[3] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.
[4] A. Sultana, L. Zhao, and X. Fernando, "Efficient resource allocation in device-to-device communication using cognitive radio technology," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10024–10034, Nov. 2017.
[5] H. Meshgi, D. Zhao, and R. Zheng, "Optimal resource allocation in multicast device-to-device communications underlaying LTE networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8357–8371, Sep. 2017.
[6] Z. Yang, N. Huang, H. Xu, Y. Pan, Y. Li, and M. Chen, "Downlink resource allocation and power control for device-to-device communication underlaying cellular networks," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1449–1452, Jul. 2016.
[7] N. Cheng *et al.*, "Performance analysis of vehicular device-to-device underlay communication," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5409–5421, Jun. 2017.
[8] R. Lan, W. Wang, A. Huang, and H. Shan, "Device-to-device offloading with proactive caching in mobile cellular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.
[9] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6300–6410, 2017.
[10] S. Soleimani and X. Tao, "Cooperative crossing cache placement in cache-enabled device to device-aided cellular networks," *Appl. Sci.*, vol. 8, no. 9, pp. 1–14, 2018.
[11] Y. Zhao and W. Song, "Energy-aware incentivized data dissemination via wireless D2D communications with weighted social communities," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 945–957, Dec. 2018.
[12] K. Zhu, W. Zhi, L. Zhang, X. Chen, and X. Fu, "Social-aware incentivized caching for D2D communications," *IEEE Access*, vol. 4, pp. 7585–7593, 2016.
[13] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.
[14] D. Malak, M. Al-Shalash, and J. G. Andrews, "Optimizing content caching to maximize the density of successful receptions in device-to-device networking," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4365–4380, Oct. 2016.
[15] W. Song and Y. Zhao, "Efficient interference-aware D2D pairing for collaborative data dissemination," in *Proc. IEEE Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–6.
[16] Y. Zhao, W. Song, and W. Zhuang, "Stable device pairing for collaborative data dissemination with device-to-device communications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1251–1264, Apr. 2018.
[17] L. Miao, B. Bai, and W. Chen, "4-DMWM approach for caching based optimal D2D pairing and channel allocation: Centralized and distributed algorithm design," *IEEE Access*, vol. 4, pp. 9213–9224, 2016.
[18] B. Bai, L. Wang, Z. Han, W. Chen, and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: A hypergraph framework," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 74–81, Aug. 2016.
[19] H. Zhang, L. Song, and Z. Han, "Radio resource allocation for device-to-device underlay communication using hypergraph theory," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4852–4861, Jul. 2016.
[20] J. Jiang, S. Zhang, B. Li, and B. Li, "Maximized cellular traffic offloading via device-to-device content sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 82–91, Jan. 2016.
[21] Z. Su, Q. Xu, F. Hou, Q. Yang, and Q. Qi, "Edge caching for layered video contents in mobile social networks," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2210–2221, Oct. 2017.
[22] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.
[23] W. Song, "Analysis of a distance-based pairing scheme for collaborative content distribution via device-to-device communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9245–9256, Sep. 2019.
[24] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.
[25] 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.
[26] F. Wang, L. Song, Z. Han, Q. Zhao, and X. Wang, "Joint scheduling and resource allocation for device-to-device underlay communication," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, 2013, pp. 134–139.
[27] J. Gu, S. J. Bae, S. F. Hasan, and M. Y. Chung, "Heuristic algorithm for proportional fair scheduling in D2D-cellular systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 769–780, Jan. 2016.
[28] H. Xing and W. Song, "Collaborative content distribution with an end-to-end caching framework," *IEEE Access*, vol. 8, pp. 54345–54360, 2020.
[29] H. Planatscher and M. Schober. (Jan. 2015). *SCPSolver—An Easy to Use Java Linear Programming Interface*. [Online]. Available: http://scpsolver.org/
[30] P. Kleinschmidt and H. Schannath, "A strongly polynomial algorithm for the transportation problem," *Math. Program.*, vol. 68, no. 1, pp. 1–13, 1995.
[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[32] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proc. Nat. Acad. Sci.*, vol. 99, pp. 2566–2572, Feb. 2002.
[33] C. Jarray and A. Giovanidis, "The effects of mobility on the hit performance of cached D2D networks," in *Proc. 14th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Tempe, AZ, USA, 2016, pp. 1–8.
[34] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?" *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 536–550, Apr. 2013.
[35] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," 2019. [Online]. Available: arXiv:1909.07972.
[36] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.

WEI SONG (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2007. In 2009, she joined the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, where she is currently an Associate Professor. Her current research interests include Internet of Things, mobile edge computing, mobile crowdsensing, and device-to-device communications. She received the Best Paper Award from the 2018 IEEE ICC, the 2014 UNB Merit Award, the Best Student Paper Award from the 2013 IEEE CCNC, the Top 10% Award from the 2009 IEEE MMSP, and the Best Paper Award from the 2007 IEEE WCNC. She is the Chair of the Joint Computer and Communications Chapter of IEEE New Brunswick Section. She has co-chaired tracks/symposiums for IEEE VTC Fall 2010, IWCMC 2011, IEEE GLOBECOM 2011, IEEE ICC 2014, IEEE VTC Fall 2016, and IEEE VTC Fall 2017.

HAORU XING received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017, and the master's degree in computer science from the University of New Brunswick, Fredericton, NB, Canada, in 2019. Her research interests include device-to-device communications, mobile edge computing, and video content distribution.