IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 26, 2019, accepted April 12, 2019, date of publication April 23, 2019, date of current version May 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912674

Sociality and Mobility-Based Caching Strategy for Device-to-Device Communications Underlying Heterogeneous Networks

GUANJIE SHAN AND QI ZHU

Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding author: Qi Zhu (zhuqi@njupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61571234 and Grant 61802155.

ABSTRACT Proactive content caching at the wireless network edge, such as users and small base stations (SBSs), is an effective way to deal with high mobile traffic. In this paper, based on the user mobility and social relationships, we investigate the optimal caching strategy in device-to-device (D2D) communications underlying heterogeneous networks, where several SBSs are within the coverage of a macrobase station (MBS). Except for SBSs, important users (IUs) hired by an operator also cache files. First, assuming that user preference, i.e., the probability distribution of different file requests of users, are unknown, we cluster users and predict each user preference based on their history file requests by fitting to the Zipf distribution. Second, we derive the closed-form expression of the average system cost by jointly considering the mobility of users and the social relationship between them. With the purpose of minimizing the average system cost, we optimize the SBSs and IU caching strategies. The optimization problem is NP-complete. To solve the problem, we demonstrate that this problem belongs to the minimization of a supermodular function over a partition matroid, and thus, we provide a locally greedy caching algorithm with an approximation ratio of 2 to obtain the sub-optimal solution in polynomial time. Finally, since the operator can reduce the system cost by hiring more IUs, requiring higher payments, we reach a tradeoff by determining the number of IUs. The simulation results show that the proposed caching strategy outperforms the traditional caching strategy, and the suboptimal solution obtained by proposing the greedy algorithm is close to the optimal solution.

INDEX TERMS Caching, mobility, partition matroid, social relationship, supermodular function.

I. INTRODUCTION

According to a recent study published by Cisco, data traffic will increase exponentially from 2016 to 2021 [1], accounting for 63% of the total Internet traffic. Locally caching popular files is a key technology to meet the huge demand of data traffic. 5G heterogeneous networks relieve the traffic load of MBS by deploying SBSs. However, the backhauls of SBSs will become a bottleneck of system performance. Caching technology enables SBSs and IUs to proactively cache popular files locally. When users request cached files, they can receive them directly from the SBSs or IUs, instead of occupying the backhaul of SBSs or bandwidth of MBS when receiving them. Caching technology avoids network congestion in traffic peak periods and reduces the delay, thus improving the quality of service (QoS) [2].

Compared with content libraries, the cache capacity of SBSs and IUs is relatively small, so it is necessary to develop a proper caching strategy to increase the cache hit ratio. There have been some studies on caching in wireless networks. In [3], [4], and [5], the users' social properties are utilized when developing the caching strategy. In [3], the social ties, jointly with physical distance, are used as a factor that affects the caching cost, and a social-aware non-cooperative game is established to minimize the total cost of the whole network by incentivizing selfish users to cache data for others. In [4], based on the users' information in the social network, users are divided into different sets and a user can receive the object

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

file with a lower price from users in the same set. The goal of this paper is to minimize content provisioning costs. Social properties are also considered in [5] to select important users to cache contents, thus formulating a many-to-one matching game of choosing important users, and a many-to-many matching game of deciding caching files. However, these three papers do not take the user mobility into account and assume that the user preference is known, which is unrealistic.

There are studies that provide caching strategy for scenarios with mobility, such as [6], [7], and [8]. In [6], the user mobility are considered, which are modeled by a Markov chain, and an optimization problem of minimizing the amount of data downloaded from MBS is formulated. An optimal distributed caching policy or a distributed greedy caching policy is proposed to solve the problem depending on delay deadline. Quer *et al.* [7] used a Markov chain to model the user mobility in a D2D underlying heterogeneous network, and optimized the users' and SBSs' caching strategies to minimize the system cost. However, standard linear integer programming tools are used to solve the optimization problem when there are a lot of users, and the complexity of solution is very high. These two studies do not consider unknown file popularities or the effect of social properties. Zhang *et al.* [8] jointly considered the social properties and mobility of users, while predicting the cache hit ratio, and designed a greedy algorithm to obtain a caching strategy aimed at maximizing the caching gain. However, the scenario is not a heterogeneous network that will be widely used in 5G, so the proposed caching strategy may not apply to heterogeneous networks that are more complicated. The authors in [9], [10], [11], [28] and [29] all considered caching strategies for scenarios where the content popularities or user preferences are unknown. Reference [9] formulated the problem of maximizing utility, which is the inverse of delay, and the contents are clustered to reduce complexity. Based on regret learning at a small cell base station and cloud, the content popularity is learned to aid the cache and update content. The aim of the study in [10] is to minimize the service delay. To develop a caching strategy, users are grouped by cluster analysis and reinforcement learning is used to learn the content popularity. In [11], the optimization problem of maximizing the total expected reward is demonstrated to be a knapsack problem if the popularity of the files is known. Then the popularity is learned by applying a multi-armed bandit model based on demand history, and thus the optimal caching strategy can be obtained. The authors in [28] presented TrendCaching, a novel caching replacement method that optimizes cache performance based on learning the unknown popularity of video contend. Müller *et al.* [29] proposed a novel algorithm, which can learn context-specific content popularity by regularly observing context information of the scene, for context-award proactive caching. However, these papers only focus on predicting the popularity and do not consider the user mobility and social properties. In addition, Golrezaei *et al.* [27] utilized the caching helpers, such as femtocells, to enhance the performance of caching. A distributed caching problem was formalized and approximation algorithms are proposed to solve this problem. Leonardi and Neglia [12] optimized the overall performance in a dense cellular network where base stations have limited-size cache. A class of simple and fully distributed caching policies was introduced to solve this problem and achieved excellent performance. But these two papers don't consider that hiring users to caching files in D2D enabled scenarios can further improve the performance of caching, and also does not consider the mobility and sociality of users either.

However, mobility and sociality is of great importance when we determine the caching strategy. Caching technology has been successfully applied in wired networks, but there are some challenges when applying it to wireless network. One of the most important reasons causing these challenges is the user mobility in wireless network. A caching strategy that does not consider mobility may perform poorly in mobile wireless network. For example, in the current time slot, the SBS caches a file which is high likely to be request by a user in its coverage, but in the next time slot, this user may leave this SBS, and a new user who is high likely to request another file may enter the coverage of this SBS. As a result, the cache hit ratio will be drop duo to the mobility. As to sociality, it may also affect the establishment of D2D connection. In real life, accepting files sent by others can be very dangerous because these files may contain viruses. For security reasons, people usually only accept files sent by familiar people. If we do not consider sociality and cache files in a user whose neighbors are not familiar with him, even if the physical condition of the communication holds, caching files in him is invalid.

Considering users' mobility and social relationship among them, this paper predicts user preference based on their history requests and studies caching strategies of SBSs and IUs to minimize system cost in a D2D communication underlying heterogeneous network consisting of a MBS and several SBSs. The main contributions of this paper are summarized as follow.

- We use K-means to cluster users into different types according to their history file requests based on the assumption that users with similar interests have basically the same file preferences. Then we can obtain each type's empirical probability distribution of requesting different files based on history file requests. Since this probability distribution is inaccurate when the historical data is limited, it is fitted to the Zipf distribution, which is widely used to describe content popularity or user preference and provides a more accurate predicted probability distribution.
- We derive the probabilities of three different ways for users to receive their requested files in the next time slot, which are from the IUs, SBSs and MBS, based on the user preference, user mobility, social relationship between users, and caching strategy of the IUs and SBSs. The average system cost of the next time slot based on probability theory is then derived, and thus

the optimization problem for minimizing the average system cost, which is a nonlinear integer programming problem, is obtained.

- The optimization problem is NP-complete. We first provide the method to obtain the optimal solution by substituting variables. However, the complexity of this method is too high so a suboptimal solution with lower complexity is required. We demonstrate that the objective function of this problem is a monotonous supermodular function, and the constraints can be regarded as a partition matroid. On this basis, we provide a polynomial time greedy algorithm with a ratio of 2 to obtain suboptimal solution.

The remainder of the paper is organized as follows: in Section II we introduce the system model; In Section III we propose the prediction algorithm of user preference; In Section IV, the optimization problem of minimizing average system cost is formulated, and the locally greedy caching algorithm is proposed to obtain suboptimal solution; In Section V we provide the method of determining the number of IUs; In Section VI the simulation results are presented to show the performance of the proposed algorithm. In section VII we concludes the paper.

II. SYSTEM MODEL

We consider a scenario as shown in Fig. 1. There is one MBS, which has the whole file library, in the network. There are U users and S SBSs in the coverage of MBS. Users can communicate with each other through D2D. User $u \in \mathcal{U} = \{1, 2, \ldots, U\}$ can only cache Vu files. Each SBS $s \in \mathcal{S} = \{1, \ldots, S\}$ has the same cache capacity V_{SBS}. The coverage of small base stations may overlap. In reality, file can be classified into different classes, such as sports program and military program and so on. So we divide file library into C classes and each class $c \in \mathcal{C} = \{1, \ldots, C\}$ contains F_c files. Thus the whole file library is a set of $F = C * F_c$ files and we denote it with $\mathcal{F} = \{1, \ldots, C * F_c\}$.

We assume that all the files have the same size. The minimum communication time for downloading a file through D2D or small base station is $t_{\min}$ and $t_{\min\prime}$ respectively.

Time is divided into slots with a discrete index $t \in \mathbb{N}$, and all time slots have the same duration which is T. The duration usually lasts for several minutes. The start time of time slot t is τ_t, and we can get $\tau_{t+1} - \tau_t = T$ since the duration of one time slot is T. At the beginning of each time slot, the MBS can know the information of whether the distance between users meets the requirements of D2D communication, i.e., users' initial D2D connection situation $\mathcal{D}^t = \{d_{i,j}^t : i = 1, \ldots, U; j = 1, \ldots, U\}$ in the current slot, where $d_{i,j}^t$ is 1 if user i and user j can communicate through D2D and is 0 otherwise. In each time slot, each user randomly requests one file according to their preferences, and these U requests constitute file request vector $\mathcal{R}^t = \{r_u^t : u = 1, \ldots, U\}$, where $r_u^t \in \mathcal{F}$ is the file requested by user u in time slot t. To simplify the model, we assume that each user requests a file at the beginning of a time slot.

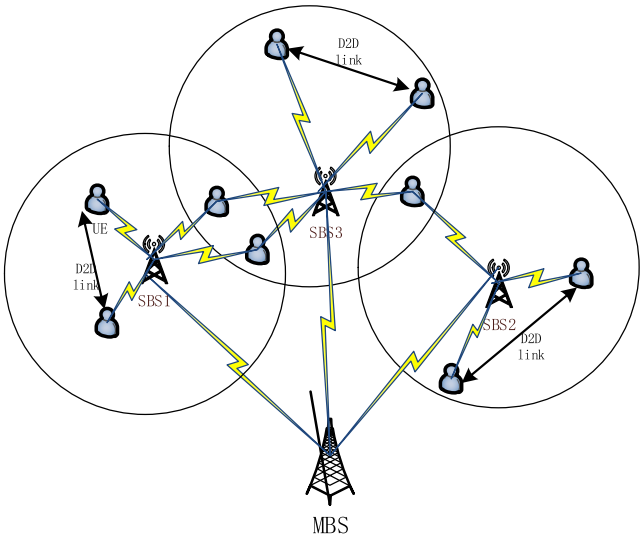

FIGURE 1. D2D communication underlying heterogeneous network.

There are three ways for users to obtain request file, from the caches of the important users around him through D2D communication, from the caches of the SBSs connecting to him, from the MBS, respectively. Like [7], we use ξ_1, ξ_2, ξ_3 to represent the system cost of these three ways respectively and they satisfy $\xi_1 < \xi_2 < \xi_3$. The system cost here can be regard as proportion of downloading delays or consumed bandwidths or battery usages through these three ways. After receiving the user's file request, IUs and SBSs decide whether to send the requested file to the user randomly according to the mobility between them. For example, when the speed of an IU relative to the request user is fast, the IU is more unwilling to send the requested file to request user because he has to stop moving and wait for the completion of requested file's transmitting. In the current time slot, MBS infers $\mathcal{D}^{t+1}$ based on the known $\mathcal{D}^t$, then synthetically considers the mobility and social relationship of users to determine the optimal caching strategy for the next time slot, and places the files that need to be cached at IUs and SBSs.

D2D communication among users is mainly affected by mobility. If two users are in close proximity, i.e., the distance of these two users is less than a certain distance, their mobile devices will be connected through WiFi or Bluetooth direct in a D2D manner. It's worth noting that the "certain" distances of different pairs of users are different, and is related to channel fading, transmit power and so on. However, because of users' mobility, the physical distance between users is always changing. However, only within a certain distance can D2D communication be established. Therefore, whether physical relationship exists between two users, i.e. the physical distance between them is within the maximal D2D communication distance, can be considered as a probabilistic problem [8]. If there is a physical relationship between two users, we call them connected and the time of connection is referred as connection time. The time between two adjacent connections is referred as interval time.

To model users' mobility, we assume that the connection time and the interval time obey the exponential distribution [8]. In addition, because users only can communicate with SBSs when they are within the coverage of SBSs, and the relative distance between the user and the SBS can change due to the mobility of the user although the location of the SBSs is fixed, similar to D2D communication, we can also use exponential distribution to model the connection time and interval time between the user and the SBS.

Let $PC^t_{i,j}$ indicates the physical relationship between user i and user j in time slot t, and $PC^t_{i,j} = 1$ if they are connected and $PC^t_{i,j} = 0$ otherwise. According to probability theory [13], continuous-time Markov chain can be used to model the connection between users. If we know the connection situation between user i and user j at time t_0, the probability of user i and user j being connected at time t_c can be obtained by

$$P(PC^{t_c}_{i,j} = 1|PC^{t_0}_{i,j}) = \begin{cases} \dfrac{\lambda_{i,j} - \lambda_{i,j}e^{-(\lambda_{i,j}+\mu_{i,j})(t_c-t_0)}}{\lambda_{i,j}+\mu_{i,j}}, & if\ PC^{t_0}_{i,j} = 0 \\ \dfrac{\lambda_{i,j} + \mu_{i,j}e^{-(\lambda_{i,j}+\mu_{i,j})(t_c-t_0)}}{\lambda_{i,j}+\mu_{i,j}}, & if\ PC^{t_0}_{i,j} = 1 \end{cases} \quad (1)$$

where $\mu_{i,j}$ and $\lambda_{i,j}$ are the exponential distribution parameters of the connection time and interval time between user i and user j respectively.

Similarly, we assume that the connection time and the interval time between user u and SBS s obey the exponential distributions with parameters $\mu'_{u,s}$ and $\lambda'_{u,s}$, respectively. Indicator variable $PD^t_{u,s}$ represents the physical relationship between user u and SBS s. If we know the connection situation at time t_0, then we can obtain the probability that user u and SBS s are connected at t_c time by

$$P(PD^{t_c}_{u,s} = 1|PD^{t_0}_{u,s}) = \begin{cases} \dfrac{\lambda'_{u,s} - \lambda'_{u,s}e^{-(\lambda'_{u,s}+\mu'_{u,s})(t_c-t_0)}}{\lambda'_{u,s}+\mu'_{u,s}}, & if\ PD^{t_0}_{u,s} = 0 \\ \dfrac{\lambda'_{u,s} + \mu'_{u,s}e^{-(\lambda'_{u,s}+\mu'_{u,s})(t_c-t_0)}}{\lambda'_{u,s}+\mu'_{u,s}}, & if\ PD^{t_0}_{u,s} = 1 \end{cases} \quad (2)$$

Based on security considerations, the successful establishment of D2D communication in this paper also involves social relationship. Only users with close social relationship are willing to establish D2D communication. Let $S_{i,j}$ denotes social closeness between user i and user j, using the Adamic/Adar method in [14], we can calculate it according to the Adamic/Adar method in [14], we can calculate it based on their social attributes by

$$S_{i,j} = \sum_{k \in A_i \cap A_j} \frac{1}{\log(frequency(k))} \quad (3)$$

where A_i denotes the social attributes of user i, the social attributes are the public information or tags posted by users on social network, such as the groups that a user join, the city a user lives in, a user's net friends, hobbies and interests of a user and so on. Since these social attributes are public and don't involve user privacy, they are not sensitive and the operator can obtain them by cooperating with social network service providers. $frequency(k)$ denotes the number of users that have social attribute k. The implication of expression (3) is as follows. If user i and user j have a common social attribute which few people have, this social attribute can be more able to explain the social closeness of these two users. We denote social closeness threshold with S_T, and social relationship indicator with $s_{i,j}$. There is a social relationship between user i and user j when $S_{i,j} >= S_T$ and $s_{i,j} = 1$. Otherwise, there is no social relationship between them and $s_{i,j} = 0$. We describe social relationship among users with graph $G_s(VU, E_s)$, where VU is the set of users and E_s is the social relationship among them.

Caching files in a users' device occupies storage space of the device. Because of the selfish nature of users, users are reluctant to cache files. Only IUs hired by operators will act as cache nodes. We introduces the concept of social importance to help operator to select IUs:

$$\theta_u = \alpha \cdot V_u + \beta \cdot B_u, \quad u = 1, \ldots, U \quad (4)$$

where V_u and B_u are cache capacity and betweenness centrality respectively. α and β are weight coefficients and $\alpha + \beta = 1$. Betweenness centrality can be calculated by [14]:

$$B_u = \sum_{i=1}^{U-1} \sum_{j=i+1}^{U} \frac{b_{i,j}(g_u)}{b_{i,j}}, \quad (5)$$

where $b_{i,j}$ denotes the number of shortest paths between vertex $i, j \in VU$ in graph G_s. $b_{i,j}(g_u)$ denotes the number of shortest paths passing user u.

Ranking users in descending order by social importance, the operator will select the top N users as IUs when needing N IUs. In the following, IU_n represents the n-th IU, i.e. the user who has the n-th biggest social importance.

The main symbols and notations are summarized in Tab. I. It is worth noting that this table does not list all the symbols and notations that appear in this paper. Some symbols for temporary use, such as those introduced to simplify derivation and proof, are not included in this table.

III. PREDICTION OF USER PREFERENCE

In our scenario, we assume that each user's file preference is unknown, and MBS only has history file requests $\mathcal{H} = \{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H_U}\}$ during the past T_b time slots, where $\mathbf{H_u} = (h_{u,1}, h_{u,2}, \ldots, h_{u,T_b})$ is the history file requests of user u, and $h_{u,t_b} \in \mathbf{H_u}$ is the file requested by u in the t_b-th time slot of former T_b time slots. According to the history file requests $\mathcal{H}$, the initial empirical probability of user u requesting files from class c can be calculated based on request times by:

$$\hat{P}_{u,c} = \frac{\sum_{t_b=1}^{T_b} \mathbf{1_A}(h_{u,t_b} \in c)}{T_b}, \quad u \in \mathcal{U},\ c \in \mathcal{C} \quad (6)$$

TABLE 1. Main notations.

Symbol	Definition
$u, U, \mathcal{U}$	User index, total number of users, user set.
$s, S, \mathcal{S}$	SBS index, the total number of SBSs, SBS set.
$c, C, \mathcal{C}$	Class index, total number of classes, class set.
$f, F_c, F, \mathcal{F}$	File index, number of files in one class, total number of files, file library.
V_u, V_{SBS}	Cache capacity of user u and SBS respectively.
$t_{\min}, t_{\min}'$	Minimum communication time for downloading one file through D2D and SBS respectively.
t, τ_t, T	Time slot index, start time of time slot t, the duration of each time slot.
$d_{i,j}^t$	The indicator that shows whether user i, j can communication through D2D in time slot t.
r_u^t	The file that user u requests in time slot t.
ξ_1, ξ_2, ξ_3	The cost of obtaining a file from IUs, SBSs, and MBS respectively.
$PC_{i,j}^t$	The indicator that shows the physical relationship between user i and user j in time slot t.
$PD_{u,s}^t$	The indicator that shows the physical relationship between user u and SBS s in time slot t.
$S_{i,j}, s_{i,j}$	Social closeness and social relationship indicator between user i and user j respectively.
IU_n	The n-th IU.
$\hat{P}_{u,f}$	The predicted probability of user u requesting f.

where $\mathbf{1}_{\mathbf{A}}(x)$ is indicator function, and $\mathbf{1}_{\mathbf{A}}(x) = 1$ if x is true and $\mathbf{1}_{\mathbf{A}}(x) = 0$ otherwise.

In real life, users can be divided into different types. For example, some users like science fiction movies best, and some users like comedy programs best. That is to say, users of the same type can be considered to have basically the same probability distribution [7]. If we can accurately determine the number of user types and the users included in each user type, not only can we reduce the number of categories of user preference, but also increase the number of each user's history file requests because all users belonging to one type can be treated as one user. It makes the empirical probability distribution more accurate, which is conducive to the further prediction of user preference.

We use the K-means [15] to classify user types. But before using K-means, the value of K should be confirmed first. We use the Gap Statistic method [16] to determine K.

The process of Gap Statistic method is as follows: First, K-means algorithm is applied to different K, and K clustering centers $M_k, k = 1, \ldots, K$ and K clusters $O_k, k = 1, \ldots, K$ are obtained after each clustering is completed. Then the sum of the distances from all data points to their clustering centers under the current K value is calculated as a measure of the current model, which is denoted as D_K.

$$D_K = \sum_{k=1}^{K} \sum_{X \in O_k} \|X - M_k\|, \tag{7}$$

where $X \in O_k$ represents a data point belonging to cluster O_k.

Afterwards we define $Gap(K) = \mathrm{E}(\log D_K) - \log D_K$ as Gap Statistic. $\mathrm{E}(\log D_K)$ is the expectation of $\log D_K$. Monte Carlo simulation is usually used to generate this value. Finally, The best K is the K that maximizes $Gap(K)$.

After determining the optimal K value, we use this K value to perform K-means algorithm and thus obtaining K clustering centers which denote empirical probability distribution of the K types of users. Then the probability distribution is used to predict user preference.

It is the fact that a user is most likely to request the several files he is most interested in, and the more interested he is in a file, the more likely he is to request it. So the user preference fits well with the characteristics of Zipf distribution, which has been used in some references, such as [17], [18], to describe user preference. Therefore, we fit the empirical probability distribution acquired by K-means to Zipf distribution. The Zipf distribution is as follows:

$$P_c = \frac{(1/rank(c))^s}{\sum_{i=1}^{C} (1/c)^s}, \quad c = 1, 2, \ldots, C \tag{8}$$

where P_c denotes the probability that a user requests files belong to c-th file class, $rank(c) \in \{1, \ldots, C\}$ represents the rank of c-th file class for this user,s is the parameter of Zipf distribution, which describes the skewness of user preference.

We can see that Zipf distribution is decided by s. Therefore fitting the empirical probability distribution to Zipf distribution only needs to determine the corresponding s.

By the logarithm of both sides of (8) [19], we can obtain:

$$\mathrm{In}(P_c) = -s\mathrm{In}(rank(c)) - \mathrm{In}\left(\sum_{c=1}^{C} (1/c)^s\right), \quad c = 1, 2, \ldots, C \tag{9}$$

It can be seen that the logarithm of the probability of each file class' being requested is linearly related to the logarithm of the rank of this class, with a slope of $-s$ and an intercept of $-\mathrm{In}(\sum_{c=1}^{C} (1/c)^s)$. The top-ranking class in Zipf distribution occupies the majority of requests, so we only consider the five largest request probabilities in the empirical probability distribution. After carrying out linear regression analysis on the logarithm of both request probability and rank, we obtain the parameters of Zipf distribution, and then calculate the request probability of each class according to the rank in the empirical probability distribution. Then, assuming that the probability distribution of requesting files belonging to the same class obeys uniform distribution, the predicted user preference can be obtained: $\mathcal{P}_{\mathcal{Z}} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_U\}$, where $\hat{\mathbf{P}}_u = (\hat{P}_{u,1}, \ldots, \hat{P}_{u,1*F_c}, \ldots, \hat{P}_{u,C*F_c}), \quad u = 1, \ldots, U$ is the preference of user u, and $\hat{P}_{u,f}$ denotes the probability of user u requesting file f, which can be calculated by:

$$\hat{P}_{u,f} = \frac{\hat{P}_{u,c_f}^{C}}{F_{c_f}}, \quad f = 1, 2, \ldots, F \tag{10}$$

where c_f is the class that file f belongs to and satisfies $\sum_{i=1}^{c_f-1} F_i + 1 \le f \le \sum_{i=1}^{c_f} F_i$, $\hat{P}^C_{u,c_f}$ denotes the probability of user u requesting files of class c_f by fitting Zipf distribution.

Algorithm 1 User Preference Prediction Algorithm

Input: history file request $\mathcal{H} = \{\mathbf{H_1}, \mathbf{H}_2, \ldots, \mathbf{H_U}\}$
Output: predicted user preference $\mathcal{P}_\mathcal{Z}$
1: Calculate the initial empirical probability distribution according to (6)
2: **for** $K = 1 : U$ **do**
3: Calculate D_K according to (7)
4: Calculate $\mathrm{E}(\log D_K)$ by Monte Carlo Simulation
5: Calculate $Gap(K) = \mathrm{E}(\log D_K) - \log D_K$
6: **end for**
7: $optK = \arg\max(Gap(K))$
8: According to the initial empirical probability distribution, use K-means algorithm to cluster users into $optK$ types, and obtain each type's users set O_K and clustering center M_K, where $K \in \{1, \ldots, optK\}$
9: **for** $K = 1 : optK$ **do**
10: Sort M_K in descending order, thus obtain the result M_K^{sorted} and the ranking vectors of users belong to K-th user type: $R_K = \{r_{k,1}, \ldots, r_{k,C}\}$, where $r_{k,c}$ is the rank of c-th file class for K-th user type
11: $\mathbf{x} = \{\log i : i = 1, \ldots, 5\}$, $\mathbf{y} = \{\log M_K^{sorted}(i) : i = 1, \ldots, 5\}$
12: Conduct linear regression on $(\mathbf{x}, \mathbf{y})$ them and get s
13: **for** $c = 1{:}C$ **do**
14: Calculate K-th user type's file class preference $P_{K,c}, \quad c = 1, \ldots, C$ according to (8)
15: Let $\hat{P}^C_{u,c} = P_{K,c}, \forall u \in O_K$
16: According to (10), calculate the predicted user preference $\hat{\mathbf{P}}_u, \forall u \in O_K$
17: **end for**
18: **end for**
19: $\mathcal{P}_\mathcal{Z} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_U\}$

User preference prediction algorithm is given in Algorithm 1. The input of the algorithm is the users' history file request. Based on this, the initial empirical probability distribution of each file class is obtained. Then, the number of user types is calculated by Gap Statistics to cluster users. Finally, the user preference is predicted by fitting to Zipf distribution.

IV. THE OPTIMIZATION PROBLEM OF MINIMIZING AVERAGE SYSTEM COST

From the operator's point of view, every penny it pays to hire IUs will bring the greatest benefit to the system. This benefit comes from reducing the system cost by hiring IUs to cache files. However, even if the IUs are determined, because capacities of IUs and SBSs are limited, different caching strategy leads to different system cost, thus the operator still has to develop proper caching strategy of IUs and SBSs to minimize the system cost under this selection of IUs. Therefore, before the operator selects IUs, it must first be able to determine the caching strategy that can minimizing the system cost when the selection of IUs is given. It should be noted that because of the social importance, selecting IUs means choosing users with the greatest social importance as IUs. This section first discusses how to optimize caching strategy to minimum average system cost when selecting IUs.

A. PROBLEM FORMULATION

The system cost for all users to receive the requested files in time slot is

$$\xi(t) = \sum_{u \in \mathcal{U}} \xi_u(t) \tag{11}$$

where $\xi_u(t)$ is the cost for user u to receive the requested file in time slot t, and it is

$$\xi_u(t) = \begin{cases} \xi_1, & if \ \ case1 \\ \xi_2, & if \ \ \neg\,(case1) \wedge \ \ case2 \\ \xi_3, & if \ \ \neg\,(case1 \vee case2) \wedge case3 \end{cases} \tag{12}$$

where, $case1$ denotes that user u acquires files from IUs through D2D communication; $case2$ denotes that user u acquires files from SBSs; $case3$ denotes that user u acquires files from the MBS;

Because the caching strategy cannot be changed in current time slot t and the user file request has been determined, the system cost $\xi(t)$ is determined. What we need to do is optimizing the caching strategy in time slot $t+1$ to minimize the average system cost $\mathrm{E}(\xi(t+1))$ based on the D2D connections between users, user preference, mobility and sociality of users in time slot t. We substitute $\mathrm{E}(\xi)$ for $\mathrm{E}(\xi(t+1))$ to simplify the notation in the following. The average system cost is as follows:

$$\begin{aligned} \mathrm{E}(\xi) &= \sum_{u \in \mathcal{U}} \mathrm{E}(\xi_u) \\ &= \sum_{u \in \mathcal{U}} [\xi_1 P(case1) + \xi_2 P(\neg\,(case1) \wedge \ \ case2) \\ &\quad + \xi_3 P(\neg\,(case1 \vee case2) \wedge case3)] \\ &= \sum_{u \in \mathcal{U}} [\xi_1 P(case1) + \xi_2 (1 - P(case1)) P(case2) \\ &\quad + \xi_3 (1 - P(case1))(1 - P(case2)) P(case3)]. \end{aligned} \tag{13}$$

Using the law of total probability, we have

$$\begin{aligned} \mathrm{E}(\xi) = \sum_{u \in \mathcal{U}} \sum_{f \in \mathcal{F}} P\left(r_u^{t+1} = f\right) \Big[& \xi_1 P\left(case1 | r_u^{t+1} = f\right) \\ & + \xi_2 \left(1 - P\left(case1 | r_u^{t+1} = f\right)\right) P\left(case2 | r_u^{t+1} = f\right) \\ & + \xi_3 \left(1 - P\left(case1 | r_u^{t+1} = f\right)\right) \\ & \times \left(1 - P\left(case2 | r_u^{t+1} = f\right)\right) \\ & * P\left(case3 | r_u^{t+1} = f\right) \Big] \end{aligned} \tag{14}$$

where $P\left(r_u^{t+1}=f\right)$ denotes the probability of user u requesting file f, and it is equal to $\hat{P}_{u,f}$, which can be obtained through user preference prediction algorithm.

We assuming that the operator hires N IUs in time slot $t+1$, and the caching strategy of IUs is $\mathcal{X}_{IU}^{t+1}=\left\{\mathbf{X}_{IU_1}^{t+1},\mathbf{X}_{IU_2}^{t+1},\ldots,\mathbf{X}_{IU_N}^{t+1}\right\}$, where $\mathbf{X}_{IU_n}^{t+1}=(x_{IU_n,1}^{t+1},x_{IU_n,2}^{t+1},\ldots,x_{IU_n,F}^{t+1})$ is the caching strategy vector of n-th IU and $x_{IU_n,f}^{t+1}$ is a binary variable which equals 1 when n-th IU caches file f and equals 0 otherwise.

We first derive the probability of receiving request file from IUs when user u requests file f, i.e. $P\left(case1|r_u^{t+1}=f\right)$.We have that

$$\begin{aligned}
&P\left(case1|r_u^{t+1}=f\right)\\
&=P\left(\mathrm{A}_{u,f,1}\vee\ldots\vee\mathrm{A}_{u,f,N}|r_u^{t+1}=f\right)\\
&=1-P\left(\neg\mathrm{A}_{u,f,1}\wedge\neg\mathrm{A}_{u,f,2}\wedge\ldots\wedge\neg\mathrm{A}_{u,f,N}|r_u^{t+1}=f\right)\\
&=1-\prod_{n=1}^{N}\left(1-P\left(\mathrm{A}_{u,f,n}|r_u^{t+1}=f\right)\right) \qquad (15)
\end{aligned}$$

where event $\mathrm{A}_{u,f,n}$ indicates that user u can obtain the requested file from the n-th IU. The first equality holds because as long as one IU is willing to transmit the file to user u, the user can obtain the request file from IUs through D2D communication, i.e. satisfying *case*1. The third equality holds because the willingness of different IUs to transmit files to the request user is independent of each other.

Assuming that the probability of n-th IU's being willing to transmit request file to user u is $P\left(t_{u,n}^{d2d}\geq t_{\min}\right)$, where $t_{u,n}^{d2d}$is the duration of connection time between user u and n-th IU and $t_{\min}$is the minimum communication time for downloading each file through D2D, we have that

$$\begin{aligned}
&P\left(\mathrm{A}_{u,f,n}|r_u^{t+1}=f\right)\\
&=P\left(PC_{u,IU_n}^{\tau_{t+1}}=1,t_{u,n}^{d2d}\geq t_{\min},s_{u,IU_n}=1,x_{IU_n,f}^{t+1}\right.\\
&\left.=1|r_u^{t+1}=f,PC_{u,IU_n}^{\tau_t}\right)\\
&=P\left(PC_{u,IU_n}^{\tau_{t+1}}=1|PC_{u,IU_n}^{\tau_t}\right)P\left(t_{u,n}^{d2d}\geq t_{\min}\right)\\
&\quad\times P\left(s_{u,IU_n}=1\right)*P\left(x_{IU_n,f}^{t+1}=1\right)\\
&=P\left(PC_{u,IU_n}^{\tau_{t+1}}=1|PC_{u,IU_n}^{\tau_t}\right)\left(\int_{t_{\min}}^{+\infty}\mu_{u,IU_n}e^{-\mu_{u,IU_n}t}dt\right)\\
&\quad\times s_{u,IU_n}x_{IU_n,f}^{t+1}\\
&=P\left(PC_{u,IU_n}^{\tau_{t+1}}=1|PC_{u,IU_n}^{\tau_t}\right)e^{-\mu_{u,IU_n}t_{\min}}s_{u,IU_n}x_{IU_n,f}^{t+1}. \qquad (16)
\end{aligned}$$

The first equality holds because $PC_{u,IU_n}^{\tau_t}$ is known in time slot t and event $\mathrm{A}_{u,f,n}$ is equal to that there is physical relationship as well as social relationship between n-th IU and user u at the beginning of time slot $t+1$, and the n-th IU caches file f and the n-th IU is willing to transmit request file to user u. The second equality holds because these events can be considered independent of each other and $P\left(PC_{u,IU_n}^{\tau_{t+1}}=1|PC_{u,IU_n}^{\tau_t}\right)$ can be calculated by (1). We substitute $Pd2d_{u,n}^{t+1}$ for $P\left(PC_{u,IU_n}^{\tau_{t+1}}=1|PC_{u,IU_n}^{\tau_t}\right)e^{-\mu_{u,IU_n}t_{\min}}s_{u,IU_n}$ to simplify the notation and get

$$P\left(\mathrm{A}_{u,f,n}|r_u^{t+1}=f\right)=Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}. \qquad (17)$$

We substitute (17) into (15) and obtain

$$P\left(case1|r_u^{t+1}=f\right)=1-\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right). \qquad (18)$$

Similar to IUs' caching strategy, we denote caching strategy of SBSs with $\mathcal{X}_{SBS}^{t+1}=\{\mathbf{X}_{SBS_1}^{t+1},\mathbf{X}_{SBS_2}^{t+1},\ldots,\mathbf{X}_{SBS_S}^{t+1}\}$, where $\mathbf{X}_{SBS_s}^{t+1}=(x_{SBS_s,1}^{t+1},x_{SBS_s,2}^{t+1},\ldots,x_{SBS_s,F}^{t+1})$ is the caching strategy of SBS s. In the same way, we can get

$$P\left(case2|r_u^{t+1}=f\right)=1-\prod_{s=1}^{S}\left(1-Psbs_{u,s}^{t+1}x_{SBS_s,f}^{t+1}\right) \qquad (19)$$

where $Psbs_{u,s}^{t+1}=P\left(PD_{u,s}^{\tau_{t+1}}=1|PD_{u,s}^{\tau_t}\right)e^{-\mu_{u,s}t_{\min}}$. Noticing that whether users can communicate with SBSs doesn't take social relationship into account, the expression of $Psbs_{u,s}^{t+1}$ does not contain indicative variable that represent social relationship compared to $Pd2d_{u,n}^{t+1}$.

Because users can communicate with MBS all the time and MBS has the whole file library, so we have

$$P\left(case3|r_u^{t+1}=f\right)=1. \qquad (20)$$

Substituting (18)-(20) into (14), we can get

$$\begin{aligned}
\mathrm{E}(\xi)=&\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}}\hat{P}_{u,f}\left[\xi_1\left(1-\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right)\right)\right.\\
&+\xi_2\left(\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right)\right)\\
&\times\left(1-\prod_{s=1}^{S}\left(1-Psbs_{u,s}^{t+1}x_{SBS_s,f}^{t+1}\right)\right)\\
&+\xi_3\left(\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right)\right)\\
&\left.\left(\prod_{s=1}^{S}\left(1-Psbs_{u,s}^{t+1}x_{SBS_s,f}^{t+1}\right)\right)\right]\\
=&\sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f}\left[\xi_1+(\xi_2-\xi_1)\right.\\
&\times\left(\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right)\right)+(\xi_3-\xi_2)\\
&\times\left(\prod_{n=1}^{N}\left(1-Pd2d_{u,n}^{t+1}x_{IU_n,f}^{t+1}\right)\right)\\
&\left.\left(\prod_{s=1}^{S}\left(1-Psbs_{u,s}^{t+1}x_{SBS_s,f}^{t+1}\right)\right)\right]. \qquad (21)
\end{aligned}$$

The goal is to minimize the system cost by optimizing the caching strategy, so the optimization problem can be expressed as

$$\begin{aligned}\min_{x_{SBS,f}^{t+1},x_{IU_n,f}^{t+1}} \quad & \mathrm{E}(\xi) \\ \text{s.t.} \quad & \sum_{f=1}^{F} x_{SBS_s,f}^{t+1} \le \mathrm{V}_{SBS}, \quad s=1,2,\ldots,S \\ & \sum_{f=1}^{F} x_{IU_n,f}^{t+1} \le \mathrm{V}_{IU_n}, \quad n=1,2,\ldots,N \\ & x_{SBS_s,f}^{t+1} \in \{0,1\}, \quad x_{IU_n,f}^{t+1} \in \{0,1\} \end{aligned} \tag{22}$$

where the first constraint is the cache capacity limit of all SBSs, and the second constraint is the cache capacity limit of IUs.

B. PROBLEM ANALYSIS AND SOLUTION

From (21), we can see that this problem is a pseudo-Boolean optimization problem whose objective function is a pseudo-Boolean function. Therefore, this problem is a nonlinear integer programming problem which is NP-complete [20]. But the existing integer programming tools can only solve linear integer programming. In [20] and [21], a method is provided to replace the product term of two binary variables x_1 and x_2 in the nonlinear integer programming with a new binary variable. This method only needs to substitute $x_{1,2}$ for x_1x_2 in the objective function, and then add four constraints: (1) $x_{1,2} \ge 0$; (2) $x_{1,2} \le x_1$; (3) $x_{1,2} \le x_2$; (4) $x_{1,2} \ge x_1 + x_2 - 1$. Based on this method, we give a method to replace a product term of k binary variables with a new binary variable.

Lemma 1: By adding $k+2$ constraints: (1) $x_{1,\ldots,k} \ge 0$; (2) $x_{1,\ldots,k} \le x_1; \ldots;$ (k + 1) $x_{1,\ldots,k} \le x_k$; (k + 2) $x_{1,\ldots,k} \ge x_1+,\ldots,+x_k-(k-1)$, the product term of k binary variables $x_1, x_2, \ldots, x_k, k \ge 2$ can be replaced by a new binary variable $x_{1,2,\ldots,k}$ in the nonlinear integer programming.

Proof:

(i) When $k = 2$, according to [20], we can know that *Lemma 1* holds.

(ii) Assuming that *Lemma 1* holds when $k = n$, then we can get the following inequalities: (1) $x_{1,\ldots,n} \ge 0$; (2) $x_{1,\ldots,n} \le x_1; \ldots;$ (n + 1) $x_{1,\ldots,n} \le x_n$; (n + 2) $x_{1,\ldots,n} \ge x_1 + x_2+,\ldots,+x_n-(n-1)$.

(iii) When $k = n+1$, we have to add four constraints (1) $x_{1,\ldots,n+1} \ge 0$; (2) $x_{1,\ldots,n+1} \le x_{1,\ldots,n}$; (3) $x_{1,\ldots,n+1} \le x_{n+1}$;(4) $x_{1,\ldots,n+1} \ge x_{1,\ldots,n} + x_{n+1} - 1$ to substitute $x_{1,\ldots,n+1} = x_1x_2\ldots x_{n+1} = x_{1,\ldots,n}x_{n+1}$ for the product term of $x_{1,\ldots,n}$ and x_{n+1}. According to inequalities (2)-(n + 1) in (ii), we know that constraint (2) in (iii) can be changed to n constraints which are $x_{1,\ldots,n+1} \le x_1; \ldots; x_{1,\ldots,n+1} \le x_n$; Combining the inequality (n + 2) in (ii) with the constraint (4) here and we can get $x_{1,\ldots,n+1} \ge x_{1,\ldots,n} + x_{n+1} - 1 \ge x_1 + \ldots + x_n + x_{n+1} - n$. Therefor the four constraints here can be changed into $n+3$ constraints which are (1)$x_{1,\ldots,n+1} \ge 0$; (2) $x_{1,\ldots,n+1} \le x_1; \ldots;$ (n+1) $x_{1,\ldots,n+1} \le x_{n+1}$; (n+3) $x_{1,\ldots,n+1} \ge x_1+,\ldots,+x_{n+1}-n$. *Lemma 1* holds in this case.

To sum up, *Lemma 1* hold when $k \ge 2$, i.e. *Lemma 1* is proved. ■

By using the conclusion of *lemma 1*, the nonlinear integer programming problem (22) can be transformed into a linear integer programming problem. However, due to the need to add new binary variables, the complexity may be very high. Exactly speaking, because all the monomials containing the product term of two or more variables in the expansion of (21) need to use a new variable to substitute the product term, this method makes the number of variables change from $(N+S)F$ to $\left(2^N + 2^{N+S} - 2\right)F$. The worst-case time complexity of linear integer programming algorithm is non-polynomial time, and the number of variables increases exponentially with N. When N and F is large, the solution complexity is very high, which makes it basically impossible to solve problem (22). Therefore, suboptimal algorithm needs to be explored.

Supermodular function is often used in pseudo-Boolean optimization. When the objective function is monotone supermodular function, the upper bound of suboptimal value obtained by properly designed greedy algorithm will achieve good results [21].

Definition 1: If Ω is a finite set and $f : 2^{\Omega} \to \mathbb{R}$ is a set function, where 2^{Ω} is the power set of Ω. If this function satisfies the following condition:

$$f(\Omega_1) + f(\Omega_2) \le f(\Omega_1 \cup \Omega_2) + f(\Omega_1 \cap \Omega_2). \tag{23}$$

where Ω_1, Ω_2 is the subset of Ω. Then function f is a supermodular function.

Reference [20] provides a method for distinguishing whether a pseudo-Boolean function is supermodular, as described in Proposition 1 below:

Proposition 1: A pseudo-Boolean function is supermodular if and only if all of its second derivatives are nonnegative:

$$\Delta_{ij}(\mathbf{x}) =^{def} \frac{\partial f}{\partial x_i \partial x_j}(\mathbf{x}) \ge 0. \tag{24}$$

Lemma 2: The objective function in question (22) is a monotone decreasing supermoduler function.

Proof: Let $\mathbf{x} = \{x_{IU_1,1}^{t+1}, \ldots, x_{IU_1,F}^{t+1}, \ldots, x_{IU_N,F}^{t+1}, x_{SBS_1,1}^{t+1}, \ldots, x_{SBS_S,F}^{t+1}\}$ be the caching strategy vector of all IUs and SBSs, then the objective function in question (22) can be regarded as a function of $\mathbf{x}$:

$$\begin{aligned} f(\mathbf{x}) == \sum_{u\in U}\sum_{f\in F} \hat{P}_{u,f} \Bigg[& \xi_1 + (\xi_2 - \xi_1) \\ & \times \left(\prod_{n=1}^{N} \left(1 - Pd2d_{u,n}^{t+1} x_{IU_n,f}^{t+1}\right)\right) + (\xi_3 - \xi_2) \\ & \times \left(\prod_{n=1}^{N} \left(1 - Pd2d_{u,n}^{t+1} x_{IU_n,f}^{t+1}\right)\right) \\ & \times \left(\prod_{s=1}^{S} \left(1 - Psbs_{u,s}^{t+1} x_{SBS_s,f}^{t+1}\right)\right)\Bigg]. \end{aligned} \tag{25}$$

where we know $0 \le \hat{P}_{u,f}, Pd2d_{u,n}^{t+1}, Psbs_{u,s}^{t+1} \le 1$ according to their expressions. We uniformly express $x_{IU_n,f}^{t+1}, x_{SBS_s,f}^{t+1}$ as $x_{k,f}^{t+1}$ for the convenience of proof. When $1 \le k \le N$, $x_{k,f}^{t+1}$ represents $x_{IU_k,f}^{t+1}$; When $N+1 \le k \le N+S$, $x_{k,f}^{t+1}$ represent $x_{SBS_{k-N},f}^{t+1}$. We uniformly express $Pd2d_{u,n}^{t+1}, Psbs_{u,s}^{t+1}$ as $Pall_{u,k}^{t+1}$, When $1 \le k \le N$, $Pall_{u,k}^{t+1}$ represents $Pd2d_{u,k}^{t+1}$; When $N+1 \le k \le N+S$, $Pall_{u,k}^{t+1}$ represents $Psbs_{u,k-N}^{t+1}$. Then (25) can be transformed into

$$f(\mathbf{x}) = \sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f}\left[\xi_1 + (\xi_2-\xi_1) \times \left(\prod_{k=1}^{N}\left(1 - Pall_{u,k}^{t+1}x_{k,f}^{t+1}\right)\right) + (\xi_3-\xi_2) \times \left(\prod_{k=1}^{N+S}\left(1 - Pall_{u,k}^{t+1}x_{k,f}^{t+1}\right)\right)\right]. \quad (26)$$

(a) We first prove that $f(\mathbf{x})$ is a monotone decreasing function of $\mathbf{x}$. Take any variable $x_{k,f}^{t+1}$ and take its first derivative.

• When $1 \le k \le N$, the first derivative is

$$\frac{\partial f}{\partial x_{k,f}^{t+1}}(\mathbf{x}) = \sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f}\left[-(\xi_2-\xi_1)\,Pall_{u,k}^{t+1} \times \left(\prod_{\substack{k1=1\\k1\neq k}}^{N}\left(1 - Pall_{u,k1}^{t+1}x_{k1,f}^{t+1}\right)\right) - (\xi_3-\xi_2)\,Pall_{u,k}^{t+1} \times \left(\prod_{\substack{k2=1\\k2\neq k}}^{N+S}\left(1 - Pall_{u,k2}^{t+1}x_{k2,f}^{t+1}\right)\right)\right]. \quad (27)$$

Because $\xi_1 < \xi_2 < \xi_3$, we have $\xi_2 - \xi_1 > 0, \xi_3 - \xi_2 > 0$. Because $0 \le Pall_{u,k}^{t+1} \le 1, \forall u, \forall k$, so $0 \le \prod_{\substack{k1=1\\k1\neq k}}^{N}\left(1 - Pall_{u,k1}^{t+1}x_{k1,f}^{t+1}\right) \le 1$, $0 \le \prod_{\substack{k2=1\\k2\neq k}}^{N+S}\left(1 - Pall_{u,k2}^{t+1}x_{k2,f}^{t+1}\right) \le 1$, we can get $\frac{\partial f}{\partial y_{k,f}^{t+1}}(\mathbf{x}) \le 0$ in this case.

• When $N+1 \le k \le N+S$, the first derivative is

$$\frac{\partial f}{\partial x_{k,f}^{t+1}}(\mathbf{x}) = \sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f} \times \left[-(\xi_3-\xi_2)\,Pall_{u,k}^{t+1}\left(\prod_{\substack{k1=1\\k1\neq k}}^{N+S}\left(1 - Pall_{u,k1}^{t+1}x_{k1,f}^{t+1}\right)\right)\right] \quad (28)$$

Because $\xi_3 - \xi_2 > 0, 0 \le Pall_{u,k}^{t+1} \le 1, \forall u, \forall k, 0 \le \hat{P}_{u,f} \le 1$, we can get $\frac{\partial f}{\partial y_{k,f}^{t+1}}(\mathbf{x}) \le 0$ in this case.

By summing up above two cases, we have that for any variable $x_{k,f}^{t+1}\frac{\partial f}{\partial y_{k,f}^{t+1}}(\mathbf{x})$ is nonpositive, i.e. $f(\mathbf{x})$ is a monotone decreasing function of $\mathbf{x}$.

(b) Then we prove that $f(\mathbf{x})$ is a supermodular function. Take any two variables $x_{k1,f1}^{t+1}, x_{k2,f2}^{t+1}$ and take their second derivative.

• When $f1 \neq f2$, observing (26), we can see that there is no monomial in the polynomial expansion of $f(\mathbf{x})$ that has the factor $x_{k1,f1}^{t+1}x_{k2,f2}^{t+1}$, i.e. in this case $\frac{\partial f}{\partial x_{k1,f1}^{t+1}\partial x_{k2,f2}^{t+1}}(\mathbf{x}) = 0$.

• When $f1 = f2 = f$ and $k1 \in \{1,\ldots,N\}, k2 \in \{1,\ldots,N\}$, the second derivation is

$$\frac{\partial f}{\partial x_{k1,f}^{t+1}\partial x_{k2,f}^{t+1}}(\mathbf{x}) = \sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f} \left[(\xi_2-\xi_1)\,Pall_{u,k1}^{t+1}Pall_{u,k2}^{t+1}\left(\prod_{\substack{k=1\\k\neq k1,k2}}^{N}\left(1 - Pall_{u,k}^{t+1}x_{k,f}^{t+1}\right)\right) + (\xi_3-\xi_2)\,Pall_{u,k1}^{t+1}Pall_{u,k2}^{t+1} \times \left(\prod_{\substack{k=1\\k\neq k1,k2}}^{N+S}\left(1 - Pall_{u,k}^{t+1}x_{k,f}^{t+1}\right)\right)\right]. \quad (29)$$

According to analysis above we have $\xi_2 - \xi_1 > 0, \xi_3 - \xi_2 > 0, 0 \le Pall_{u,k}^{t+1} \le 1, \hat{P}_{u,f} \ge 0$, thus we get $\frac{\partial f}{\partial x_{k1,f1}^{t+1}\partial x_{k2,f2}^{t+1}}(\mathbf{x}) \ge 0$ in this case.

• When $f1 = f2 = f$ and $k1$ or $k2 \in \{N+1,\ldots,N+S\}$, the second derivation is

$$\frac{\partial f}{\partial x_{k1,f}^{t+1}\partial x_{k2,f}^{t+1}}(\mathbf{x}) = \sum_{u\in U}\sum_{f\in F}\hat{P}_{u,f}\,(\xi_3-\xi_2)\,Pall_{u,k1}^{t+1}Pall_{u,k2}^{t+1} \times \left(\prod_{\substack{k=1\\k\neq k1,k2}}^{N+S}\left(1 - Pall_{u,k}^{t+1}x_{k,f}^{t+1}\right)\right) \quad (30)$$

where $\xi_3 - \xi_2 > 0, 0 \le Pall_{u,k}^{t+1} \le 1, \hat{P}_{u,f} \ge 0$, so $\frac{\partial f}{\partial x_{k1,f}^{t+1}\partial x_{k2,f}^{t+1}}(\mathbf{x}) \ge 0$ in this case. ■

Summing up above three cases, we have $\frac{\partial f}{\partial x_{k1,f1}^{t+1}\partial x_{k2,f2}^{t+1}}(\mathbf{x}) \ge 0$ for $\forall k1, k2 \in \{1,\ldots,N+S\}, \forall f1, f2 \in \mathcal{F}$. According to proposition 1, we have that $f(\mathbf{x})$ is a supermodular function.

In summary, $f(\mathbf{x})$ is a monotone decreasing supermodular function and *Lemma 2* is proved.

Pseudo-Boolean optimization problems with supermodular objective functions can be classified into different types based on their different constraints. Greedy algorithms used

to obtain sub-optimal solution are also different. Here we give a concept to distinguish different types of pseudo-Boolean optimization problems which is matroids.

Definition 2: A matroid is a set system $(E, \mathcal{L})$, where E is a nonempty finite set(called the ground sets) and $\mathcal{L}$ is a family of subsets of E, i.e. $\mathcal{L} \subseteq 2^E$, (called the independent sets) and $\emptyset \in \mathcal{L}$, with the following properties:

(a) $\mathcal{L}$ has augmentation property: if $A \in \mathcal{L}, B \in \mathcal{L}$, and $|A| < |B|$, then there exists $x \in B \backslash A$ such that $A \cup \{x\} \in \mathcal{L}$.

(b) $\mathcal{L}$ has hereditary property: if $B \in \mathcal{L}, A \subseteq B$, then $A \in \mathcal{L}$. B is called an independent subset of E, thus any subset of B is also an independent subset of A.

The type of a pseudo-Boolean optimization problem is judged by the type of matroid corresponding to the constraints of this problem. We first give the definition of partition matroids and then prove that constraints in problem (22) correspond to a partition matroid.

Definition 3: $E = \bigcup_{i=1}^{k} E_i$ is the disjoint union of k sets, $l_1, \ldots, l_k$ are positive integers, and

$$\mathcal{L} = \left\{ L : L = \bigcup_{i=1}^{k} L_i, L_i \subseteq E_i, |L_i| \leq l_i \ for \ i = 1, \ldots, k \right\} \tag{31}$$

Then we call that $(E, \mathcal{L})$ is a partition matroid.

Lemma 3: Constraints in problem (22) correspond to a partition matroid.

Proof: Let $EF = \bigcup_{i=1}^{N+S} EF_i$, When $i \in \{1, \ldots, N\}, EF_i = \{1, \ldots, F\}$ is the ground set of i-th IU which represents the files he can cache; When $i \in \{N+1, \ldots, N+S\}$, $EF_i = \{1, \ldots, F\}$ is the ground set of SBS $i - N$ which represents the files SBS $i - N$ can cache. Let

$$\mathcal{LF} = \left\{ LF : LF = \bigcup_{i=1}^{N+S} LF_i, where \ LF_i \subseteq EF_i, \right. \\ \left. |LF_i| \leq \mathrm{V'}_i \ for \ i = 1, \ldots, N+S \right\}, \tag{32}$$

where $\mathrm{V'}_i$ represents the cache capacity of IUs or SBSs, and when $i \in \{1, \ldots, N\}, \mathrm{V'}_i = \mathrm{V}_{IU_n}$; when $i \in \{N+1, \ldots, N+S\}, \mathrm{V'}_i = \mathrm{V}_{SBS_{i-N}}$. In addition, $LF_i = \left\{ f : x_{i,f}^{t+1} = 1, f = 1, \ldots, F \right\}$ is the caching strategies of IUs and SBSs which satisfy the constraints in problem (22), thus $\mathcal{LF}$ is the set of all feasible caching strategies. According to the definition 3, we have that $(EF, \mathcal{LF})$ is a partition matroid and *Lemma 3* is proved. ■

Based on the above discussion, problem (22) belongs to the problem of minimizing the supermodular function defined over a partition matroid. To solve this problem, [22] showed that locally greedy algorithm can achieve at most 2 times the optimal objective function value. Of course, this is the result when the size of the input matroid is very large, and when the size is small, this algorithm can achieve much better results [20], [22], [23]. So we use locally greedy algorithm to obtain suboptimal solution of problem (22).

Algorithm 2 Locally Greedy Caching Algorithm(LGCA)

Input: $\hat{P}_{u,f}, Pall_{u,k}^{t+1}, \mathrm{V'}_i \ (i = 1, 2, \ldots, N+S)$
Output: Sub-optimal caching strategy vector $\mathbf{x}_{subopt}$

1: $\mathbf{x}_{subopt} = \mathbf{O}_{1,(N+S)F}$
2: **for** $i = N+1, N+2, \ldots, N+S, 1, \ldots, N$ **do**
3: $\quad j = 0, \mathcal{F}_{left} = \{1, \ldots, F\}$
4: $\quad$ **while** $j < \mathrm{V'}_i$ **do**
5: $\quad\quad f_{opt} \leftarrow \arg\max_{f \in \mathcal{F}_{left}} \left[f\left(\mathbf{x}_{subopt}\right) - f\left(\mathbf{x}_{subopt} | \mathbf{x}_{subopt}\left[(i-1)F + f\right] = 1\right) \right.$
6: $\quad\quad \mathbf{x}_{subopt}\left[(i-1)F + fopt\right] = 1$
7: $\quad\quad \mathcal{F}_{left} = \mathcal{F}_{left} \backslash \left\{f_{opt}\right\}$
8: $\quad\quad j = j + 1$
9: $\quad$ **end while**
10: **end for**

As shown in algorithm 2, Firstly, we initialize the caching strategy of all IUs and SBSs as a row vectors containing $1 \times (N+S)F$ elements. The for-loop from step 2 to 10 determines the caching strategy of one IU each time. In step 2 i is assigned in that order for the convenience of determining the number of IUs, which we be explained in the following. In Step 3, j represents the number of files that have been determined to place in the cache, and $\mathcal{F}_{left}$ represents the remaining set of optional files. Step 5 is to find out caching which file will reduce the value of objective function at the maximum degree, where $f\left(\mathbf{x}_{subopt} | \mathbf{x}_{subopt}\left[(i-1)F + f\right] = 1\right)$ represents what is $f\left(\mathbf{x}_{subopt}\right)$ equal to if the $\left[(i-1)F + f\right]$-th element in $\mathbf{x}_{subopt}$ changes from 0 to 1.

Now we analyze the complexity of the algorithm. Each instruction in the while-loop executes $\mathrm{V'}_i$ times. The running time of each instruction is constant except for the fifth instruction. The complexity of fifth instruction is $O(F)$ because it traverses all f to find out the file that reduces the objective function most. That is to say, the total running time of steps 4 to 9 is $O(\mathrm{V'}_i(F+1)) = O(\mathrm{V'}_i F)$. Steps 3 and while-loop are executed once in each for-loop. Step 3 requires constant running time. The running time of while loops is $O(\mathrm{V'}_i F)$, so the running time of each for-loop is $O(\mathrm{V'}_i F)$, too. For-loop need to be executed $N+S$ times, thus the total complexity is

$$O\left(F \sum_{i=1}^{N+S} \mathrm{V'}_i\right) = O(F\mathrm{V}_0) \tag{33}$$

where $\mathrm{V}_0 = \sum_{i=1}^{N+S} \mathrm{V'}_i$ is the sum of cache capacities of all IUs and SBSs. In addition to for-loop, step 1 requires constant running time, so the running time complexity of the whole algorithm is $O(F\mathrm{V}_0)$, which is polynomial time. The sub-optimal value obtained by LGCA is closest to the optimal value in polynomial time [22].

V. THE SELECTION OF IMPORTANT USERS

After solving the problem of minimizing the average system cost while selecting N IUs, the next problem is how to decide N, i.e. how to decide how many IUs should be select to act as cache nodes. We consider this problem from the operator's point of view. System benefit per unit cost (SBPUC) is defined to measure the benefits to operators while selecting N important users. Its expression is

$$G_N(t+1) = \frac{(1-\varepsilon)\,\xi_0(t+1) - \xi_s^N(t+1)}{NH} \tag{34}$$

where $\xi_s^N(t)$ is the sub-optimal value of the average system cost when selecting N IUs at time slot $t+1$ by using LGCA. $\xi_0(t+1)$ is the sub-optimal value of the average system cost when only SBSs cache files at time slot $t+1$. $0 \le \varepsilon \le 1$ is the ratio of system cost that the operator hopes to reduce by hiring IUs, compared to the cost without hiring IUs, i.e. the operator wants the average system cost be smaller than $(1-\varepsilon)\,\xi_0(t+1)$ by hiring IUs. H is the price of hiring an IU. It is worth noting that the maximal value of N is U, since the operator may hire all users as IUs in order to reduce the system cost as much as possible.

The implication of SBPUC is as follows: because caching will consume storage of user devices, the operators need to pay IUs cache files because of users' selfish nature; at the same time, hiring IUs will reduce the system cost. In order to meet the basic requirements of QoS or system upgrade, compared with only small base station caching files, the operator has a minimum expected reduction of average system cost. If the average system cost is lower, this is the additional benefit of hiring IUs to the operator. To achieve a trade-off between the cost of hiring IUs and the additional benefit, we define the ratio of extra benefit to the cost of hiring IUs as the SBPUC to help select IUs. Obviously, we need the N, which is the number of selected IUs, that maximizes $G_N(t+1)$. Let this N be $N_{opt}(t+1)$, we have

$$N_{opt}(t+1) = \arg\max\left(G_N(t+1), N = 1, 2, \ldots, U\right). \tag{35}$$

It's worth noting that we don't have to calculate $G_N(t+1)$ U times to find $N_{opt}(t+1)$. From the expression of $G_N(t+1)$ we can see that different values of N only affect $\xi_s^N(t+1)$ and NH. NH is easy to calculate but $\xi_s^N(t+1)$ need to be calculated by LGCA. Recalling the LGCA, in which we determine the caching strategy of SBSs and N IUs in turn, we can only execute LGCA once when $N = U$ to obtain all $G_N(t+1)$. The method is to let N be U in LGCA, then add several instructions between step 9 and step 10 which are: *if* $i = U+S$, then $\xi_0 = f\left(\mathbf{x}_{subopt}\right)$; *elseif* $i \in \{1, \ldots, U\}$, then $G_i = \frac{(1-\varepsilon)\xi_0 - f\left(\mathbf{x}_{subopt}\right)}{iH}$, $CS(i) = \mathbf{x}_{subopt}\left(1 : \left(S + N_{opt}\right)F\right)$ and then add an instruction after step 10 which is $N_{opt} = \arg\max_{i=1,\ldots,U}(G_i)$. N_{opt} is the number of IUs the operator should select in time slot $t+1$ and $CS(N_{opt})$ is the caching strategy of SBSs and N_{opt} IUs.

VI. PERFORMANCE EVALUATION

The simulation scenario in this paper is one MBS with 4 SBSs and 20 users in its coverage, and users can communicate with each other through D2D. Files are divided into 15 classes and each class contains 6 files, i.e. there are 90 files in the whole file library. The cache capacity of each SBS is 30 and the cache capacity of each user is 8 or 12 or 16 with the probability of 0.3, 0.6 and 0.1. The duration of each time slot is 300 seconds. The cost of receiving files from IUs, SBSs and MBS is 1, 10 and 100, respectively. It takes at least 30s for the user to receive a file successfully through D2D and takes at least 60s to receive a file successfully from the SBSs. The cost of hiring each IU is 10.

The study of [8] shows that the order of magnitude of the parameter of the exponential distribution corresponding to the connection time in reality is 10 ^−3, and for interval time the order of magnitude of the parameter is 10^−4. As a result, to model physical relationship between users, we assume that $\mu_{i,j}$ and $\lambda_{i,j}$ are random numbers obeying uniform distribution $U(1.5 * 10 \wedge -3, 3.5 * 10 \wedge -3)$ and $U(6 * 10 \wedge -4, , 8 * 10 \wedge -4)$ respectively. While simulating the connection between users and SBSs, considering that for each user, there is one SBS that he often stays at, i.e. the relative movement between them is slow, and this user are less likely to stay at other SBSs, we let $\mu'_{u,s} = \mu'_1 = 10 \wedge -3, \lambda'_{u,s} = \lambda'_1 = 9 * 10 \wedge -4$ if $u \in \{(s-1)U/S + 1, \ldots, sU/S\}$ and $\mu'_{u,s} = 3 * \mu'_2 = 10 \wedge -3, \lambda'_{u,s} = \lambda'_2 = 5 * 10 \wedge -4$ otherwise. For the social relationship among users, we assumes that there are 30 user attributes, and the probability of each user possessing any one of the attributes is equal. This paper assumes that the social closeness threshold is 6.8. The weight α and β in (4) are both 0.5. ε in (34) is 0.9.

In the simulation, there are two types of users. Ranking of different file classes for these two user types are any two permutation of $\{1, \ldots, F\}$. Then the two types of user preferences obey Zipf distributions with parameters of 1.2 and 2 respectively. According to the probability distribution, users randomly select a file class, and then randomly select a file belonging to this class according to uniform distribution.

Fig. 2 shows a comparison of the system cost obtained through three different ways. From top to bottom, the first curve shows the system cost obtain by random caching, which random places file in IUs and SBSs' caches until their caches are full. The second curve shows the system cost obtained by using popularity-based caching strategy, which is a widely used caching strategy [24]–[26]. The idea is to cache the files with the greatest popularity at each caching node. In this paper, to achieve popularity-based caching strategy, after predicting the preferences of all users, we take the average of all user preferences as the global file popularity, and all IUs and SBSs place the most popular files in their caches until their caches are full. The bottom curve shows the system cost obtained by our proposed suboptimal caching strategy. We can see that the performance of random caching is the worst because it doesn't consider the impact of user

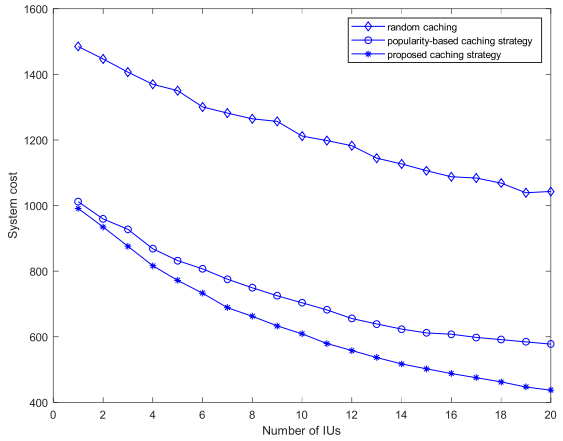

FIGURE 2. Comparison of proposed suboptimal caching strategy, popularity-based caching strategy, and random caching.

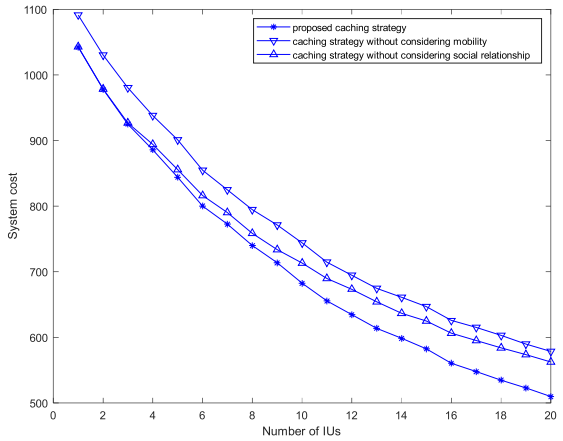

FIGURE 3. Comparison of proposed suboptimal caching strategy, caching strategy without considering mobility, and caching strategy without considering social relationship.

preferences and just caches files randomly. The system cost obtained by using this strategy is far greater than that of popularity-based caching strategy and our proposed caching strategy. The performance of popularity-based caching strategy is much better than that of random caching, but because it does not take into account the joint optimization of different IUs and SBSs, its system cost is larger than that of our proposed caching strategy, and this gap increases with the increase of the number of IUs.

Fig. 3 demonstrates the necessity of developing a caching strategy that considers mobility and sociality. There are three curves in this figure. The above curve shows the system cost by using caching strategy without considering mobility. This curve is obtained in the following way: First, remove the mobility in this paper's scenario, i.e., if a user can communicate with another user or SBS in the current time slot, they must can communicate in the next slot. Then, apply LGCA algorithm to this modified scenario to obtain the caching strategy without considering mobility, and then apply this strategy to this paper's scenario that considers mobility, and get the system cost corresponding to this strategy. We can see that because the caching strategy without considering mobility neglects the mobility in the scenario and regards the connection situation of the current time slot as the connection situation of the next time slot, the system cost obtained by using this strategy is larger than that of the proposed caching strategy. The middle curve shows the system cost by using caching strategy without considering sociality. This curve is obtained in the following way: First, remove the sociality in this paper's scenario, i.e., if two users physically meet the requirements of D2D communication, then they can establish D2D communication, regardless of whether they have social relations or not. Then apply LGCA algorithm to this modified scenario to obtain the caching strategy without considering sociality, and then apply this strategy to this paper's scenario that considers sociality, and get the corresponding system cost. Although the system cost of this strategy is basically the same as that of our proposed strategy when there are few IUs, with the increase of the number of IUs, compared with the cost of our proposed strategy, the system cost of this strategy is larger and the gap is increasing. This is because it ignores the fact that some users can't communicate with each other duo to unreliability, resulting in some files placed at the user are invalid because people around this user may not willing to communicate with him because they have no social relationship.

Fig. 4 shows a comparison of the system cost between the suboptimal caching strategy and the optimal caching strategy. The optimal caching strategy here is obtained by the method provided in Part A of Section IV. Specifically, using Lemma 1, the optimization problem in (22) can be transformed into a linear integer programming problem, and then the optimal caching strategy can be obtained by using the standard linear integer programming optimization tools to solve the problem. Because the direct solution is NP-complete, in order to reduce the computational complexity, the scenario for comparison only contains one SBS, and we only show the comparison when the number of IUs is between 1 and 4 (when the number is more than 4, the optimal value is hard or to be directly solved and may cannot be directly solved because too many variables lead to high complexity). From Fig. 4, we can see, the gap between the optimal value and the sub-optimal value is very small.

Fig. 5 shows the impact of mobility on average system cost. For simplicity, we ignore the order of magnitudes of exponential distribution parameters of connect time and interval time, which are $10 \wedge -3$ and $10 \wedge -4$, respectively. In this figure, the μ_1', λ_1' (the exponential distribution parameters of connect time and interval time between a user and the SBSs he often stays respectively) and μ_2', λ_2' (the exponential distribution parameters of connect time and interval time between a user and other SBSs) of mobility 1 are 2, 8, 4, 2.5 respectively, which means that the user is more likely to establish communication with the SBSs he often stays, and all $\mu_{i,j}$ is 3, all $\lambda_{i,j}$ is 6. These six parameters of mobility 2 are 1.66, 8.5, 3.66, 3, 2.5, 6.5, respectively. These six parameters of mobility 3 are 1.33, 9, 3.33, 3.5, 2, 7, respectively. These six parameters

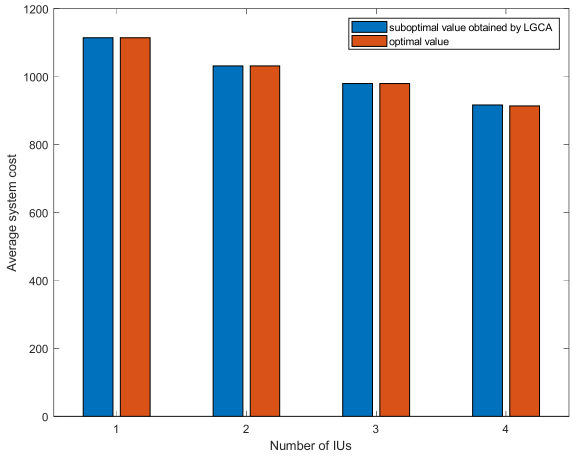

FIGURE 4. Comparison of suboptimal value and optimal value.

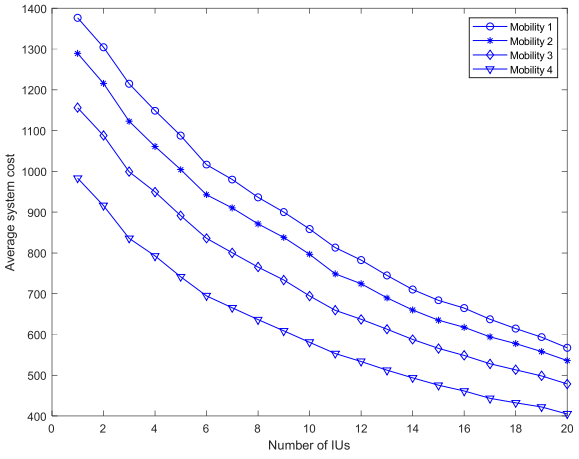

FIGURE 5. The impact of mobility on average system cost.

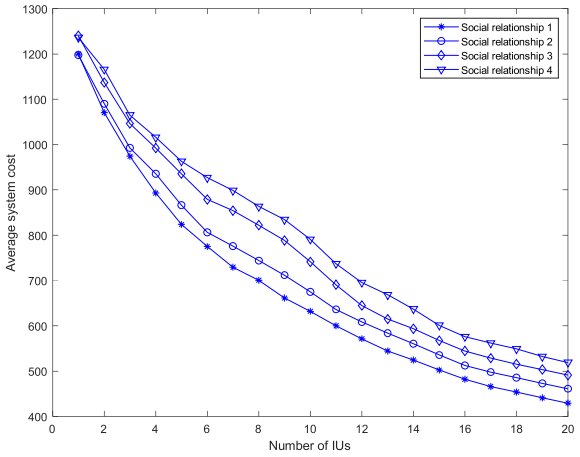

FIGURE 6. The impact of social relationship on average system cost, the social closeness thresholds of social relationship 1, social relationship 2, social relationship 3, and social relationship 4 are 6.1, 6.6, 7.1, and 7.6, respectively.

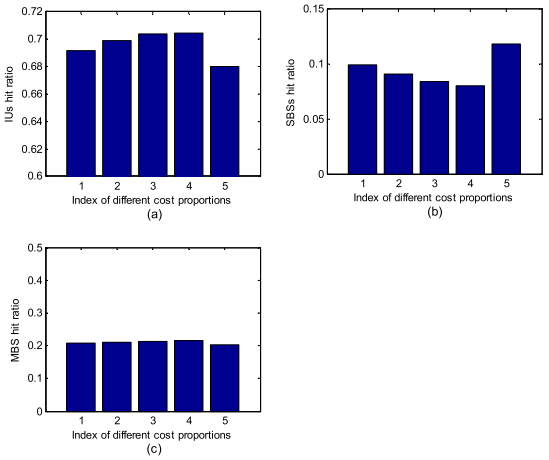

FIGURE 7. The impact of different pricing on the caching strategy. The coordinate values of the x-axis correspond to five different cost proportions: 1:10:100, 1:20:90, 1:30:70, 1:49:51, and 10:1:100, respectively.

of mobility 4 are 1, 9.5, 3, 4, 1.5, 7.5, respectively. From curve 1 to curve 4, the μ keeps decreasing while the λ keeps increasing, which means that the connect time is longer and longer and interval time is smaller and smaller both between two users and between a user and a SBS, which means that users are moving more and more slowly. Because the D2D communication among users and the connection between a user and a SBS are more and more stable, the average system cost is smaller and smaller.

Fig. 6 shows the impact of social relationship on average system cost. From curve 1 to curve 4, the social closeness thresholds are 6.1, 6.6, 7.1, 7.6, respectively, which means that the social closeness keep increasing and thus making the number of a pair of users having social relationship become smaller and smaller. Since the establishment of D2D communication requires social relationship, it is harder and harder to establish D2D communication between a user and a IU. As a result, the average system cost becomes more and more.

Fig. 7 shows the impact of different pricing, i.e. the different proportions of the costs of three ways to obtain a request file. We consider five different proportions of IUs service cost, SBSs service cost and MBS service cost, which are 1:10:100, 1:20:90, 1:30:70, 1:49:51, and 10:1:100, respectively. Among the first four proportions, (MBS service cost - IUs service cost) / (MBS service cost - SBS service cost) is increasing, which means that IUs service can reduce system cost to a greater extent than SBSs service, because the reduction of system cost is relative to the cost without caching, i.e., MBS service cost. As a result, The IUs hit rate corresponding to the first four proportions is increasing, while the SBSs hit rate is decreasing. This is in order to make better use of IUs, so that more requests are served by IUs and less requests are served by SBS, since IUs service can reduce system cost more. The fifth ratio shows another extreme case: IUs service cost is more than SBSs service cost, which means SBSs service can reduce system cost more, so the SBSs hit rate is the largest of the five and the IUs hit rate is the smallest. Note that the MBS hit rates corresponding to these five proportions are basically same. This is because the cost of MBS service is always the largest, that is to say, it always needs to minimize the MBS hit rate, so MBS hit rate is always around the minimum hit rate.

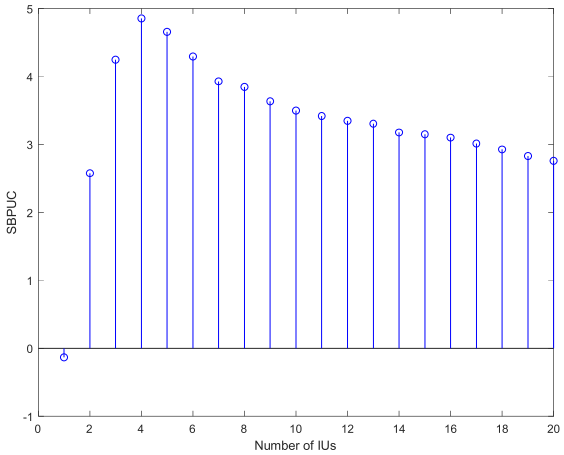

FIGURE 8. The SBPUC corresponding to different number of IUs in time slot 30.

Fig. 8 shows the SBPUC corresponding to the different number of IUs in any time slot (this figure is the time slot 30). It can be seen that SBPUC is the largest when the number is 4, so it is most appropriate to select four IUs according to social importance in the time slot 30.

VII. CONCLUSION

In this paper, we investigated how to develop the caching strategy in the D2D communication enabled heterogeneous network, while jointly considering users' mobility and social relationships, to minimize system cost. We first predicted user preference through Zipf regression based on users' history file requests. Moreover, we derived the closed-form expression of average system cost and formulated the optimization problem of minimizing average system cost. Unfortunately, this problem is NP-complete. Although the optimal solution can be obtained, the time complexity is much higher than the polynomial time. Furthermore, we proved the objective function of this problem is a supermodular function and the constraints can be mapped to a partition matroid, thus the local greedy algorithm can be used to develop suboptimal caching strategy within polynomial time. Finally, we proposed the method to decide the number of IUs. Simulation results show that the proposed caching strategy can achieve smaller system cost than traditional popularity-based caching strategy and random caching strategy. Furthermore, the performance gap between the optimal caching strategy and the sub-optimal caching strategy is very small.

REFERENCES

[1] "Cisco visual networking index: Forecast and methodology, 2016–2021," CISCO, San Jose, CA, USA, White Paper 1513879861264127, 2017.
[2] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1710–1732, 3rd Quart., 2018.
[3] 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.
[4] M. Taghizadeh, K. Micinski, C. Ofria, E. Torng, and S. Biswas, "Distributed cooperative caching in social wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1037–1053, Jun. 2013.
[5] J. Li *et al.*, "On social-aware content caching for D2D-enabled cellular networks with matching theory," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 297–310, Feb. 2019.
[6] E. Ozfatura and D. Gündüz, "Mobility and popularity-aware coded small-cell caching," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 288–291, Feb. 2018.
[7] G. Quer, I. Pappalardo, B. D. Rao, and M. Zorzi, "Proactive caching strategies in heterogeneous networks with device-to-device communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5270–5281, Aug. 2018.
[8] W. Zhang, D. Wu, X. Chen, J. Qu, and Y. Cai, "Mobility-embedded and social-aware distributed caching for D2D content sharing," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process.*, Nanjing, China, Oct. 2017, pp. 1–6.
[9] S. Tamoor-Ul-Hassan, S. Samarakoon, M. Bennis, M. Latva-Aho, and C. S. Hong, "Learning-based caching in cloud-aided wireless networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 137–140, Jan. 2018.
[10] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst.*, Barcelona, Spain, Aug. 2014, pp. 945–949.
[11] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun.*, Sydney, NSW, Australia, Jun. 2014, pp. 1897–1903.
[12] E. Leonardi and G. Neglia, "Implicit coordination of caches in small cell networks under unknown popularity profiles," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1276–1285, Jun. 2018.
[13] S. M. Ross, *Introduction to Probability Models*, New York, NY, USA: Academic, 1997.
[14] Y. Wang, M. Ding, Z. Chen, and L. Luo, "Caching placement with recommendation systems for cache-enabled mobile social networks," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2266–2269, Oct. 2017.
[15] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[16] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," *J. Roy. Stat. Soc., B (Stat. Methodol.)*, vol. 63, no. 2, pp. 411–423, Apr. 2000.
[17] X. Gan, Z. Qin, L. Fu, and X. Wang, "Unraveling the impact of users' interest on information dissemination in wireless networks," *IEEE Access*, vol. 6, pp. 32687–32699, 2018.
[18] Y. Wu, S. Yao, Y. Yang, Z. Hu, and C.-X. Wang, "Semigradient-based cooperative caching algorithm for mobile social networks," in *Proc. IEEE Global Commun. Conf.*, Washington, DC, USA, Dec. 2016, pp. 1–6.
[19] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, Nov. 2009.
[20] E. Boros and P. L. Hammer, "Pseudo-Boolean optimization," *Discrete Appl. Math.*, vol. 123, nos. 1–3, pp. 155–225, 2002.
[21] Y. Crama and P. L. Hammer, *Boolean Functions: Theory, Algorithms, and Applications (Encyclopedia of Mathematics and its Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[22] P. R. Goundan and A. S. Schulz, "Revisiting the greedy approach to submodular set function maximization," *Optim. Online*, pp. 1–25, Jul. 2007.
[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, Dec. 1978.
[24] 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.
[25] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 3619–3623.
[26] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *Proc. IEEE INFOCOM Workshops*, Orlando, FL, USA, Mar. 2012, pp. 316–321.
[27] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1107–1115.
[28] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.
[29] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024–1036, Feb. 2017.

GUANJIE SHAN received the bachelor's degree in electronic information engineering from the Nanjing University of Posts and Telecommunications, in 2017, where he is currently pursuing the master's degree in communication and information system. His research interests include application of caching in wireless networks and optimization method.

QI ZHU received the bachelor's and master's degrees in radio engineering from the Nanjing University of Posts and Telecommunications (NUPT), China, in 1986 and 1989, respectively, where she is currently a full-time Professor with the School of Telecommunication and Information Engineering. Her research interests include the technology of next-generation communication, broadband wireless access, OFDM, channel and source coding, and dynamic allocation of radio resources.