Individual Preference Aware Caching Policy Design in Wireless D2D Networks

Ming-Chun Lee, *Student Member, IEEE*, and Andreas F. Molisch, *Fellow, IEEE*
Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA, USA
Email: mingchul@usc.edu, molisch@usc.edu

***Abstract*—Cache-aided wireless device-to-device (D2D) networks allow significant throughput increase, depending on the concentration of the popularity distribution of files. Many studies assume that all users have the same preference distribution; however, this may not be true in practice. This work investigates whether and how the information about *individual* preferences can benefit cache-aided D2D networks. We examine a clustered network and derive a network utility that considers both the user distribution and channel fading effects into the analysis. We also formulate a utility maximization problem for designing caching policies. This maximization problem can be applied to optimize several important quantities, including throughput, energy efficiency (EE), cost, and hit-rate, and to solve different tradeoff problems. We provide a general approach that can solve the proposed problem under the assumption that users coordinate, then prove that the proposed approach can obtain the stationary point under a mild assumption. Using simulations of practical setups, we show that performance can improve significantly with proper exploitation of individual preferences. We also show that different types of tradeoffs exist between different performance metrics and that they can be managed through caching policy and cooperation distance designs.**

I. Introduction

Over the years, wireless data traffic has rapidly increased, consequently straining wireless networks. This trend in data traffic is expected to continue over the next several years [2]. Among all the wireless applications available today, on-demand video accounts for the largest portion of this traffic. Thus, finding an efficient approach to support this application is a paramount issue for modern wireless communication systems. Conventional approaches for increasing throughput, such as cell densification, installing large-scale antenna systems, and adopting millimeter-wave communications [3], are deemed insufficient, as such methods would entail high cost when investing in physical resources. Different from those approaches that tend to improve wireless networks without regard to the type of data to be transmitted, video caching at the wireless edge exploits the unique video accessing behavior of typical consumers and cheap storage resources to trade memory for bandwidth. In essence, different users cache different popular video files on their devices; a file request can then be satisfied either from a user's own cache or through D2D communication with a nearby user that has stored the requested file. The potential of D2D-based video caching has been widely discussed in recent years [4]–[11]. Accordingly, previously published papers have demonstrated, either in theory or in practice, that using wireless video caching with D2D communications can significantly improve throughput [8]–[11].

A. Literature Review and Motivations

Cache-aided D2D has demonstrated the ability to significantly improve network performance without the need for newly installed infrastructure and complicated coding.[1] Thus, numerous papers have been published on this topic. To understand cache-aided D2D from a theoretical point of view, [9]–[11] investigated the throughput-outage tradeoff. On the other hand, different performance metrics, including hit-rate (file outage) [9], [16], throughput [9], [17], [18], energy efficiency (EE) [18], [19], and latency [7], [39] have also been investigated in order to improve the network from different aspects. Ref. [18] particularly studied the optimizations of throughput, EE, and their tradeoff by jointly considering the effect of the cooperation distance and caching policy of D2D users. Furthermore, [19] particularly focused on battery life when proposing an energy-efficient caching policy design.

Different mathematical techniques and deployment scenarios were considered for cache-aided D2D. In [20] and [21], stochastic geometry was exploited to analyze networks and to propose caching policy designs. Considering MIMO systems, scaling laws of throughput were discussed in [22] and [23]. In [24], mobility was leveraged to enhance network performance. In [25], a randomized caching policy with a special structure for helping content delivery through D2D links is proposed and analyzed in the systems that consider millimeter-wave communications for D2D links. In [26], a cache replacement approach was proposed to accommodate the environmental dynamics. Cache-aided D2D has been a subject of great interest to many researchers, and hundreds of related papers have accordingly been published on this topic. Hence, the above literature review, by necessity, cites only a sample of

This work was supported in part by the National Science Foundation (NSF) under projects CCF-1423140 and CNS-1816699. Part of this work has been presented at the 2017 IEEE Global Communications Conference [1].

[1] Concentrated popularity distribution of video files can also be exploited in other ways, e.g., femtocaching [12]–[14] and coded multicasting [10], [15]; those approaches are outside the scope of this paper.

papers and topics. To complement our literature review, we refer to several recent survey papers [27]–[30].

Most of the existing papers for cache-aided D2D networks consider a homogeneous preference model, which assumes all users have the same file preference. In other words, each user requests files independently and randomly according to the *same* popularity distribution. However, this model is at best an approximation, because different users indeed have different tastes and preferences. Such heterogeneity in preferences of users has been observed in [31] and has also been modeled in recent works [32], [33], [37]. Furthermore, based on real-world data, results in [31] have shown that leveraging individual (user) preferences indeed can improve network performance. Thus, these abovementioned observations clarify that cache-aided D2D networks can be further improved by using a heterogeneous model, instead of a homogeneous model, in the network design. This is because designs that are based on the latter model are restricted, as it does not consider individual user preferences.

Researchers have recently begun to consider individual preferences in their analysis, and studies have accordingly shown that it is possible to use this information to improve the performance of wireless caching networks [31], [34]–[40].[2] In [31], individual preferences were studied, and a machine-learning approach was used to learn the user's preferences and accordingly decide which video should be preloaded onto a local device cache based on the preferences of that particular user. While this kind of approach (also known as the "Netflix challenge") is very important for recommendation systems and preloading on individual devices, it is not applicable to cache-aided D2D networks. In [34], an individual preference-aware weighted sum utility of users was formulated and optimized when the files were being cached at the BSs. Meanwhile, [35] designed a caching policy by assuming that users in different groups have different file preferences; the goal then is to maximize the successful file discovery probability of different groups without taking possible interference into account. In [36], a content push strategy was designed to maximize the D2D offloading gain for a particular demand realization by jointly considering the influences of user preference and sharing willingness. In [37], user preferences were used to maximize the offloading probability without accounting for the details of the physical layer. Using individual preference and user similarity, [38] proposed a caching content allocation approach to maximize a specifically defined utility. While [39] focused on estimating individual preferences using a learning-based algorithm, the study provided a caching policy that exploits the estimated preferences in order to minimize the average delay of D2D networks. Lastly, [40] proposed a game theoretical-cooperative caching design by assuming that users know exactly what files they want to request.

Despite this progress, the understanding of how individual preferences can be used to improve cache-aided D2D networks is still far from complete. It is still unclear whether integrating individual preferences into the design can improve network performance significantly. Moreover, the interplay between and among the different performance metrics, e.g., throughput, EE, and hit-rate, and the corresponding tradeoff that results from these interactions are still subject to further studies. Most importantly, the existing papers do not provide sufficient evaluations based on real-world data. Accordingly, our paper aims to address these issues.

B. Contributions

In this paper, we consider a BS-assisted cache-aided D2D network, where users can obtain the desired files from the BS, caches of neighboring users via D2D links, and their local caches. We assume that users have different preferences and caching policies; thus, our goal is to maximize network utility by designing individual preference-aware caching policies for users. We analyze the network based on the clustering and random-push scheduling [18] and then propose a non-convex utility maximization problem formulation. We then show that our proposed utility maximization problem can be applied to solve different practical problems, e.g., throughput, hit-rate, and EE optimization problems, as well as several tradeoff problems. Hence, it is sufficient to investigate a general solution approach for the proposed utility maximization problem. In addition, we discuss how the proposed utility maximization problem can be used in scenarios with different fading and user distributions.

We assume that users perfectly know the individual preferences of all the other users in the network and that they are allowed to coordinate with one another. With these considerations, we solve the utility maximization problem and obtain the users' caching policies. The idea of the proposed approach is to optimize individually and iteratively the users' caching policies until convergence. We show that this method is fairly simple to use, improves at each iteration, and converges to a stationary point under a mild assumption. We then evaluate the proposed caching policies in networks under realistic setups; in particular, we adopt the practical individual preference generator proposed in [33] based on extensive real-world data.

Our results show that network performance can significantly improve when the information on individual preferences is properly exploited. We also compare the performances of those networks that optimize throughput, EE, and hit-rate using the proposed utility maximization framework and investigate the influences of the cooperation range of the D2D network. The results indicate that there are tradeoffs between these important metrics, and we can manage the tradeoffs by properly exploiting our proposed framework. Finally, we show how the proposed designs can be used as good reference designs for obtaining effective designs in networks with complicated scheduling. We emphasize that, to the best of our knowledge, this is the first work that validates from different perspectives of the network the benefits of exploiting user preferences and that gains insights through simulations based on real-world data. To sum up, our paper makes the following contributions:

- We formulate a utility maximization problem by considering individual preferences in the analysis. Caching policies that optimize several practically important metrics,

[2] The conference version of this paper [1] is one of the earliest studies that took individual preferences into consideration.

e.g., throughput, EE, and hit-rate, and their tradeoffs, can be obtained by solving the problem. We then propose a general low-complexity approach for solving the utility maximization problem, and prove that the solution approach improves at each iteration and then converges to a stationary point.

- Considering the realistic setup based on extensive real-world data, we conduct comprehensive simulations to show the benefits of exploiting individual preferences and to demonstrate tradeoffs between different performance metrics.

C. Organization of This Paper

The remainder of the paper is organized as follows. In Sec. II, we present the network and individual preference models. We then analyze the networks and formulate the utility maximization problem in Sec. III. Also in this section, we relate the proposed problem to different practical problems and show the effects of fading and user distributions. The caching policy design is proposed and studied in Sec. IV. Extensive simulation results are provided in Sec. V. We conclude the paper in Sec. VI. The proofs and detailed derivations are relegated to the Appendices.

II. Network and Individual Preference Models

We consider a BS-assisted cache-aided wireless D2D network, where the BS helps file delivery and makes scheduling decisions. The users can obtain the desired files by retrieving them from their own caches, D2D communications, or BS links. The file library consists of M files, and for simplicity, we assume that all files have the same size.[3] Each user is able to cache S files in its storage. Besides, to have a nontrivial case, we require $S < M$. Furthermore, in most practical situations, $S << M$ will hold. Users can be either active or inactive. An active user is a user that places a request that needs to be satisfied and participates in the D2D cooperation, i.e., the user sends files to other users that request them. On the other hand, an inactive user is a user that does not place its own request but still participates in the D2D cooperation.

We consider a widely used clustering network model [9], [17], [18], [22], [23]. In the model, there is a square cell with a BS at the center point, and the cell is divided into equal-sized square clusters with side length D. The users are allowed to cooperate via D2D communications only with users in the same cluster. The "cooperation distance" or "cluster size" we henceforth reference thus corresponds to the dimension D of such a cluster, not the cell radius of the BS. We assume that there is no interference between users of different clusters; this can be achieved by letting different clusters use different time/frequency resources with "spatial reuse." We will in the following focus on a single cluster. Nevertheless, our results can easily be extended to multi-cluster scenarios. We denote the number of active users in a cluster as K_{A} and the number of inactive users in a cluster as K_{I}. The total number of users in a cluster is then $K = K_{\mathrm{A}} + K_{\mathrm{I}}$. The described model is shown in Fig. 1.

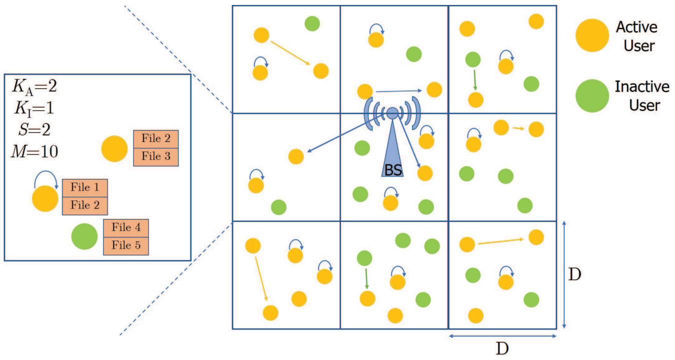


Fig. 1: This figure shows an example of the network model. In the left-middle cluster, we have $K_{\mathrm{A}} = 2$, $K_{\mathrm{I}} = 1$, $S = 2$, and $M = 10$.

We consider serving users via "random-push" scheduling [18], which functions as follows. For a cluster, the BS first randomly selects an active user without knowing whether its request can be satisfied by the user's own (local) cache. If the selected user can obtain the desired file from the local cache, i.e., the desired file is actually cached by the selected user, then the user request is satisfied immediately. Otherwise, the BS checks whether the other users in the D2D network store the desired content and whether the channel quality between the selected user and the other users storing the desired file is larger than the minimum requirement (in terms of capacity) for a D2D transmission. If yes, then a D2D link is used to transmit the desired content; otherwise, the user needs to use the BS link to access the content. We assume that the BS has an unlimited bandwidth backhaul to repositories that store all files in the library. This guarantees that any request from a selected user can always be satisfied, albeit at a potentially high cost. After scheduling for the selected user, the remaining active users check whether the files in their local caches can satisfy their requests. If yes, then their requests are satisfied. Clearly, such scheduling can guarantee that at least one user is served and all users are scheduled fairly in the sense that every user is selected with equal probability by the BS for service. We note that there exist scheduling approaches with better overall throughput, e.g., priority-push scheduling [18] and dynamic link scheduling [41]. However, using them has other drawbacks, such as unfairness and high complexity. Most importantly, it is very challenging to obtain tractable formulations for these advanced scheduling schemes [18], [41]. On the other hand, random-push scheduling leads to tractable expressions for different critical metrics and is easy to implement, thus serving as a good reference system. It should be noted that the definitions of user distribution and channel model can influence the scheduling behaviors. We will discuss them later in Sec. III.D.

We represent individual preferences for requesting video

[3] This paper generally focuses on understanding the impact of individual preferences on network performance and the tradeoffs among different performance metrics. Thus, an investigation of how to deal with heterogeneous file sizes is beyond our scope. However, based on our numerical investigations (omitted for brevity), a performance evaluation using designs with equal filesize assumption could be representative of an evaluation without equal filesize assumption.

files as probabilities. We denote the request probability of user k for file m, i.e., the probability that user k wants file m in the future, as a_m^k, where $0 \leq a_m^k \leq 1, \forall m, k$ and $\sum_{m=1}^{M} a_m^k = 1, \forall k$. Note that the preferences of inactive users are actually not used in the proposed problem formulation and solution approach later; their modeling of preferences is for the purpose of consistency. Different users can have different caching policies. As such, we denote b_m^k as the probability that user k would cache file m, whereas the caching policy of user k is described by $\{b_m^k\}_1^M$, where $0 \leq b_m^k \leq 1, \forall m, k$, and $\sum_{m=1}^{M} b_m^k \leq S, \forall k$. An implementation of this probabilistic caching policy can be found in [13]. Note that the caching policy becomes deterministic when considering the limiting case that b_k^m becomes 1 or 0. This is useful in the situation where the central controller knows a priori which users are going to be in a cluster, so that the caching policy can avoid detrimental file overlap (compare [7]). Such a deterministic predictability of user location occurs in a place where the same people are in geographical proximity every day, e.g., in an office scenario.

We assume that users perfectly know the individual preferences of the other users in the same cluster. In other words, users know $a_m^k, \forall m, k$. We also assume that users can coordinate with one another to design their caching policies for a common goal, e.g., maximizing network throughput. The users then coordinate such that they cache files by fully considering the caching policies and preferences of other users.[4] Accordingly, this approach suits scenarios where user locations are deterministic, e.g., office scenarios. Note that the information exchange and the coordination between users are assumed to be handled centrally by the BS. Moreover, it will be shown later that although users are assumed to coordinate, the algorithm does not necessarily need to be operated in a centralized manner. On the contrary, since users have a common goal and know the individual preferences of other users perfectly, each user can independently implement the same caching policy design algorithm and accordingly obtain the same coordinated design that gives the policies of all users. Consequently, each user can extract its own caching policy and then implement independently.

In this work, users can access the desired files from their local caches, from the caches of other users, and from the BS. We consider different utilities when different types of approaches are used. The utility of accessing a file via a BS link is then denoted as U_{B}, the utility of accessing a file via a D2D link as U_{D}, and the utility of accessing a file via the user's own cache as U_{S}. Although we consider all users to have the same utility, the extension to the case that different users have different utilities is straightforward. Besides, the utility can be set differently for different practical purposes, such as throughput and EE maximization, etc. However, we will generally assume that $U_{\text{B}} \leq U_{\text{D}} \leq U_{\text{S}}$, which implies that using self-access is superior to using a D2D link, and using a D2D link is superior to using a BS link. We will discuss this more thoroughly in Sec. III.C. Table I summarizes the notations frequently used in this paper.

III. Caching Policy Design Problem

Our goal here is to design caching policies that optimize network utility by using information about individual preferences. In this section, we first derive the access probabilities of different accessing approaches for a user. Based on the results, we then formulate the caching policy design problem that we aim for. To clarify the usefulness of the proposed network utility maximization problem, we then show how it can be used to solve various practical problems. Finally, we discuss how the proposed network utility can accommodate different scenarios with different fading and user distributions.

A. *Fundamental Access Probability*

Consider the system model in Sec. II. We denote $\mathcal{U}_A$ as the index set of active users and $\mathcal{U}_I$ as the index set of inactive users, and $\mathcal{U} = \mathcal{U}_A \bigcup \mathcal{U}_I$. We denote the channel between user k and user l as $h_{k,l}$ and the corresponding signal-to-noise ratio (SNR) as $\text{SNR}_{k,l}$. We let C be the minimal capacity requirement for establishing a D2D link. When user k is selected, the probability that user k accesses the desired file through a BS link is expressed as

$$P_{\text{B}}^k = \sum_{m=1}^{M} a_m^k \left[\prod_{l \in \mathcal{U}} \left(1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}\right) \right], \tag{1}$$

where $\mathbf{1}_{\{h_{k,l},C\}} = 1$ if $\log_2(1 + \text{SNR}_{k,l}) > C$; otherwise $\mathbf{1}_{\{h_{k,l},C\}} = 0$. Note that $\prod\limits_{l \in \mathcal{U}} \left(1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}\right)$ is the probability that file m can be obtained only via a BS link, and $a_m^k \prod\limits_{l \in \mathcal{U}} \left(1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}\right)$ is the probability that the user wants file m but file m can be obtained only via a BS link. We define the self-access probability, i.e., the probability that user k can obtain the desired file from its own cache, as

$$P_{\text{S}}^k = \sum_{m=1}^{M} a_m^k b_m^k. \tag{2}$$

By using P_{B}^k and P_{S}^k, the probability that user k obtains the desired file via a D2D link is

$$\begin{aligned} P_{\text{D}}^k &= 1 - P_{\text{S}}^k - P_{\text{B}}^k \\ &= 1 - \sum_{m=1}^{M} a_m^k \left[\prod_{l \in \mathcal{U}} \left(1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}\right) \right] - \sum_{m=1}^{M} a_m^k b_m^k. \end{aligned} \tag{3}$$

B. *Utility Maximization Problem Formulation*

Now, we derive the expected utility of the network. We assume that for any user k, the channel gains of all possible associated D2D links, i.e., $\mathbf{1}_{\{h_{k,l},C\}}, \forall l$, are independent (see use cases in Sec. III.D). Using the results in Sec. III.A, the

[4] A common approach used to incentivize users to coordinate is through payment by the network operator. Alternatively, since each user generally benefits from D2D communications, a token-based approach similar to the traditional file-sharing networks can ensure that specific users do not exploit the system without contributing to it. Generally, the topic of giving incentives is an important one for D2D networks. However, this is already beyond the scope of this paper, albeit it is still considered an important future direction of this paper.

TABLE I: Summary of Frequently Used Notations

Notations	Descriptions
M; S; D	Number of files in the library; number of files that can be stored by a user; cluster size
K_a; K_i; K	Number of active users; of inactive users; of total users in a cluster
$\mathcal{U}_\text{A}$; $\mathcal{U}_\text{I}$; $\mathcal{U}$	Index set of active users ; of inactive users; of all users
a^k_m; b^k_m	Request probability of user k for file m; caching probability of user k for file m
$L_{k,l}$; w_k	Probability of a successful D2D link between users k and l; weight of user k
U_B; U_D; U_S	Utility of using BS link; of using D2D link; of using self-caching
T_B; T_D; T_S	Throughput of using BS link; of using D2D link; of using self-caching
C_B; C_D; C_S	Cost of using BS link; of using D2D link; of using self-caching
U_net; T_net; C_net; H_net; EE_net	Utility; throughput; cost; energy efficiency; hit-rate of the network
P^k_B; P^k_D; P^k_S	Elementary access probabilities: refer to the clear definition of (1); of (3); of (2)

utility of the selected user k when user k is selected by the BS is expressed as

$$U_k = U_\text{D} \cdot P^k_\text{D} + U_\text{B} \cdot P^k_\text{B} + U_\text{S} \cdot P^k_\text{S}. \tag{4}$$

We denote weights $w_1, w_2, ..., w_{K_\text{A}}$ as the weighting on different users, which indicates the relative priority of users. Since users are randomly selected by the BS, the expected utility contributed by the selected users is

$$\begin{aligned} U &= \sum_{k\in\mathcal{U}_A} w_k \mathbb{E}\{U_\text{D} \cdot P^k_\text{D} + U_\text{B} \cdot P^k_\text{B} + U_\text{S} \cdot P^k_\text{S}\} \\ &= \sum_{k\in\mathcal{U}_A} \frac{w_k}{K_\text{A}} \left[U_\text{D} \cdot P^k_\text{D} + U_\text{B} \cdot P^k_\text{B} + U_\text{S} \cdot P^k_\text{S}\right] \end{aligned} \tag{5}$$

The users not selected by the BS can still check whether their desired files are cached in their local caches. As such, we can obtain additional utilities from the users' ability to satisfy their own requests. Thus, the expected utility of the network is

$$\begin{aligned} U_\text{net} &= U + U_\text{local} \\ &= U + U_\text{S} \cdot \frac{1}{K_\text{A}} \cdot \sum_{k\in\mathcal{U}_A} \sum_{l\in\mathcal{U}_A, l\neq k} \sum_{m=1}^{M} w_l a^l_m b^l_m \\ &= \sum_{k\in\mathcal{U}_A} \frac{w_k U_\text{D}}{K_\text{A}} + (U_\text{B} - U_\text{D}) \sum_{m=1}^{M} S_m \\ &\quad + (K_\text{A} U_\text{S} - U_\text{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{w_k a^k_m b^k_m}{K_\text{A}}, \end{aligned} \tag{6}$$

where $S_m = \sum_{k\in\mathcal{U}_A} \frac{w_k a_{m,k}}{K_\text{A}} \prod_{l\in\mathcal{U}} (1 - b^l_m L_{k,l})$ and $L_{k,l} = \Pr\left[\log_2(1 + \text{SNR}_{k,l}) > C\right]$. Note that the derivations for (6) are shown in Appendix A. Moreover, the computation for $L_{k,l}$ will be discussed later in detail in Sec. III.D.

Using (6), the caching policy design problem that maximizes the network utility is:

$$\begin{aligned} \max_{b^k_m, \forall k,m} \quad & U_\text{net} \\ \text{subject to} \quad & \textstyle\sum_{m=1}^{M} b^k_m \leq S, \forall k, \\ & 0 \leq b^k_m \leq 1, \forall k, m. \end{aligned} \tag{7}$$

We then have the following proposition.

Proposition 1: The optimal solution of (7) must be tight at the equality of the sum constraint, i.e., for the optimal solution $(b^k_m)^*, \forall k, m$, we have

$$\sum_{m=1}^{M} (b^k_m)^* = S, \forall k. \tag{8}$$

Proof. By (6), the first-order partial derivative of U_net is:

$$\begin{aligned} \frac{\partial U_\text{net}}{\partial b^j_m} =& -(U_\text{B} - U_\text{D}) \sum_{k\in\mathcal{U}_A} \frac{w_k a^k_m}{K_\text{A}} L_{k,j} \prod_{l\in\mathcal{U}, l\neq j} (1 - b^l_m L_{k,l}) \\ &+ \mathbf{1}_{\{j\in\mathcal{U}_\text{A}\}} (K_\text{A} U_\text{S} - U_\text{D}) \frac{w_j a^j_m}{K_\text{A}}, \forall j, m, \end{aligned} \tag{9}$$

where $\mathbf{1}_{\{j\in\mathcal{U}_\text{A}\}} = 1$ when $j \in \mathcal{U}_\text{A}$; otherwise $\mathbf{1}_{\{j\in\mathcal{U}_\text{A}\}} = 0$. Since $U_\text{B} \leq U_\text{D} \leq U_\text{S}$, $0 \leq L_{k,l} \leq 1, \forall k, l$, and $0 \leq b^k_m \leq 1, \forall k, m$, we then have $\frac{\partial U_\text{net}}{\partial b^k_m} \geq 0, \forall k, m$. Therefore, U_net is non-decreasing with respect to $b^k_m, \forall k, m$, which indicates that the optimal solution of (7) must be tight at the equality of the sum constraint. □

C. Interpretations of the Utility Maximization Problem and Its Relationship to Practice

In this subsection, we show how the utility maximization problem can be used in designing caching policies to solve various practical and important problems. In the following, we consider the equal-weight case, i.e., $w_1 = w_2 = ... = w_{K_\text{A}} = 1$, for notation convenience. The extension to other weights is straightforward.

1) Throughput Maximization Problem: Consider $U_\text{B} = T_\text{B}$, $U_\text{D} = T_\text{D}$, $U_\text{S} = T_\text{S}$, and $T_\text{B} \leq T_\text{D} \leq T_\text{S}$, where T_B is the throughput of a BS link, T_D is the throughput of a D2D link, and T_S is the throughput of self-access. The utility maximization problem then becomes the throughput maximization problem, in which the expected throughput is

$$T_\text{net} = T_\text{D} + (T_\text{B} - T_\text{D}) \sum_{m=1}^{M} S_m + (K_\text{A} T_\text{S} - T_\text{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{a^k_m b^k_m}{K_\text{A}}. \tag{10}$$

2) Cost/Power Minimization Problem: Let $U_\text{B} = -C_\text{B}$, $U_\text{D} = -C_\text{D}$, $U_\text{S} = -C_\text{S}$, and $C_\text{B} \geq C_\text{D} \geq C_\text{S}$, where C_B is the cost of a BS link, C_D is the cost of a D2D link, and C_S is the cost of self-access. The problem can then be cast as the cost minimization problem, expressed as

$$\begin{aligned} \min_{b^k_m, \forall k,m} \quad & C_\text{net} \\ \text{subject to} \quad & \textstyle\sum_{m=1}^{M} b^k_m \leq S, \forall k, 0 \leq b^k_m \leq 1, \forall k, m, \end{aligned} \tag{11}$$

where

$$C_\text{net} = C_\text{D} + (C_\text{B} - C_\text{D}) \sum_{m=1}^{M} S_m + (K_\text{A} C_\text{S} - C_\text{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{a^k_m b^k_m}{K_\text{A}}.$$

If the power consumption is considered as cost, the problem is the power minimization problem.

3) Hit-Rate Maximization Problem: Let $U_{\rm B}=0$, $U_{\rm D}=1$, and $U_{\rm S}=\frac{1}{K_{\rm A}}$. The problem then is to maximize

$$H_{\rm net}=\sum_{k\in\mathcal{U}_A}\mathbb{E}\left[P_{\rm D}^k+P_{\rm S}^k\right]=1-\sum_{m=1}^{M}S_m, \tag{12}$$

which maximizes the file hit-rate of the network.

4) Throughput–Cost Weighted Sum Problem: To attain the desired tradeoff between the different metrics, a common approach is to maximize the weighted sum/difference of the different metrics [42]. For example, considering the tradeoff between throughput and cost, we can maximize

$$w_{\rm T}T_{\rm net}-w_{\rm C}C_{\rm net}, \tag{13}$$

where $w_T\geq 0$ and $w_C\geq 0$. Such a weighted sum/difference problem is equivalent to the utility maximization problem, as we let $U_{\rm B}=w_{\rm T}T_{\rm B}-w_{\rm C}C_{\rm B}$, $U_{\rm D}=w_{\rm T}T_{\rm D}-w_{\rm C}C_{\rm D}$, and $U_{\rm S}=w_{\rm T}T_{\rm S}-w_{\rm C}C_{\rm S}$. The same concept can also be used for the throughput–hit-rate tradeoff. Also, the same concept can be applied to tradeoff between more than two objectives.

5) Efficiency Problem: In some situations, we aim to maximize the efficiency, e.g., EE (bits/Joule). The following discussions then show that the efficiency maximization problem can be addressed by solving the weighted sum problem described in Sec. III.C.4.

We consider the EE maximization problem as an example. The same concept can be used for other problems. Suppose we aim to maximize EE, which is given as

$$\begin{aligned}{\rm EE}_{\rm net}&=\frac{\text{total number of bits transmitted}}{\text{total amount of energy consumed}}\\&=\frac{\text{expected throughput of the network}}{\text{expected power consumed by the network}}=\frac{T_{\rm net}}{C_{\rm net}}.\end{aligned}$$

Then, the EE maximization problem is:

$$\begin{aligned}\max_{b_m^k,\forall k,m}\quad & {\rm EE}_{\rm net}=\tfrac{T_{\rm net}}{C_{\rm net}}\\ \text{subject to}\quad & \textstyle\sum_{m=1}^{M}b_m^k\leq S,\forall k,0\leq b_m^k\leq 1,\forall k,m.\end{aligned} \tag{14}$$

This problem is then equivalent to

$$\begin{aligned}\max_{t,b_m^k,\forall k,m}\quad & t\\ \text{subject to}\quad & \tfrac{T_{\rm net}}{C_{\rm net}}\geq t,\\ & \textstyle\sum_{m=1}^{M}b_m^k\leq S,\forall k,0\leq b_m^k\leq 1,\forall k,m,\end{aligned} \tag{15}$$

Assuming that the optimal t^* is known, then the problem in (15) is equivalent to finding the optimal policy in

$$\begin{aligned}\max_{b_m^k,\forall k,m}\quad & T_{\rm net}-t^*C_{\rm net}\\ \text{subject to}\quad & \textstyle\sum_{m=1}^{M}b_m^k\leq S,\forall k,0\leq b_m^k\leq 1,\forall k,m.\end{aligned} \tag{16}$$

In observing (16), we see clearly that we have a weighted difference problem similar to that described in Sec. III.C.4, in which $w_{\rm T}=1$ and $w_{\rm C}=t^*$. Thus, it can be cast into the utility maximization framework. Also, the optimal policy should result in $T_{\rm net}-t^*C_{\rm net}=0$.

In general, we cannot know the optimal t^* a priori; however, the aforementioned idea can still be used to solve the EE maximization problem. Suppose we have the same problem as that in (16), but we now replace t^* with t. Accordingly, we have the following interpretations: (i) if the solution results in a positive number, i.e., $T_{\rm net}-tC_{\rm net}>0$, then our solution can provide an EE larger than t; (ii) if the solution gives $T_{\rm net}-tC_{\rm net}<0$, then our solution provides an EE that is less than t, and t is not achievable. As such, by adjusting t based on the results and by solving the problem in (16) using different t, we can keep optimizing t. Thus, we improve the solution. Finally, by carefully adjusting t and by solving (16) many times, we can maximize the EE. Since the utility maximization problem is non-convex, we might not find the best t^* and the corresponding user caching policies. However, we can still obtain an effective solution by using the above approach. This technique is identical to that used for solving a quasi-convex problem [43].

D. Effects of the Statistics of Wireless Channels and User Distributions

In (6), the channel quality influences the expected utility via $L_{k,l}$. Thus, understanding the general expression of $L_{k,l}$ and its relationship to channel physics is important. In this section, we provide several useful expressions for $L_{k,l}$ and then discuss its relationship to the possible scenarios. Note that if $k=l$, then $L_{k,l}=L_{k,k}=1$. Therefore in the following, we consider $k\neq l$.

Let $d_{k,l}$ be the distance between user k and user l. The input–output relationship between users k and l then follows the general expression:

$$y_l=\sqrt{{\rm PG}(d_{k,l})s_{k,l}}h_{k,l}x_k+n_l, \tag{17}$$

where y_l is the received signal at user l; x_k is the transmit signal from user k; ${\rm PG}(d_{k,l})$ is the path gain effect (channel [power] gain averaged over small-scale and large-scale fading); $s_{k,l}$ is the shadowing power gain; $h_{k,l}$ is the small-scale fading amplitude; and n_l is the Gaussian noise with power σ_n^2. Let $E_{\rm D}$ be the transmission power of the D2D link. Using (17), the received SNR for the D2D link between users k and l is ${\rm SNR}_{k,l}=\frac{E_{\rm D}|h_{k,l}|^2s_{k,l}{\rm PG}(d_{k,l})}{\sigma_n^2}$, and therefore

$$L_{k,l}={\rm Pr}\left[|h_{k,l}|^2s_{k,l}{\rm PG}(d_{k,l})>\frac{\sigma_n^2(2^C-1)}{E_{\rm D}}\right]. \tag{18}$$

We will show later some practical examples and then demonstrate how (18) is computed using user and fading distributions. The extensions to other models are feasible by leveraging the existing results of fading [44] and distance distributions [45].

1) Case 1: Systems with effective link quality control: In modern wireless communication systems, approaches such as adaptive power control and frequency-and-antenna-diversity are used to combat fading effects in wireless channels. Thus, in systems with effective link quality control, we can assume that the D2D links between users in an area can be guaranteed, leading to $L_{k,l}=1,\forall k,l$. The exact distribution of users then becomes irrelevant in this case.

2) Case 2: Systems with deterministic path-loss and shadow fading: When users are less mobile or stationary, the joint effect of pathloss and shadow fading between users is deterministic. As a result, $s_{k,l}$ and ${\rm PG}(d_{k,l})$ are the given constants and are based on the exact locations of users. In this case, we focus on characterizing the small-scale fading. Thus,

$$L_{k,l} = \Pr\left[|h_{k,l}|^2 > \frac{\sigma_n^2(2^C-1)}{E_{\mathrm{D}} s_{k,l} \mathrm{PG}(d_{k,l})}\right], \tag{19}$$

where the closed-form expressions are attainable for commonly used fading distribution. For example, let us consider a normalized Rayleigh fading whose average power is 1; we have

$$L_{k,l} = \exp\left[\frac{-\sigma_n^2(2^C-1)}{E_{\mathrm{D}} s_{k,l} \mathrm{PG}(d_{k,l})}\right]. \tag{20}$$

Note that in this case, the distribution of users can be arbitrary but deterministic.

3) Case 3: K users uniformly distributed in a square with side length D and with shadowing and small-scale fading: Here, we use the lognormal shadowing and normalized Rayleigh fading as an example. According to results in [46], the distance d between two users independently and uniformly distributed over a square area with unit side length is described by the probability density function:

$$f_{\mathrm{sq}}(d) = \begin{cases} 2d(\pi + d^2 - 4d), & 0 \le d \le 1, \\ 2d(-2 - d^2 + 4\sqrt{d^2-1} + 2\sin^{-1}\dfrac{2-d^2}{d^2}), & 1 < d \le \sqrt{2}. \end{cases} \tag{21}$$

Then, when fixing the shadowing $s_{k,l}$, again using the property of Rayleigh fading, we have

$$\begin{aligned} L_{k,l}(s_{k,l}) &= \int_0^{\sqrt{2}D} \exp\left[-\frac{\sigma_n^2(2^C-1)}{E_{\mathrm{D}} s_{k,l} \mathrm{PG}(x)}\right] f[d=x])dx \\ &= \int_0^{\sqrt{2}} \exp\left[-\frac{\sigma_n^2(2^C-1)}{E_{\mathrm{D}} s_{k,l} \mathrm{PG}(Dx)}\right] f_{\mathrm{sq}}(x))dx. \end{aligned} \tag{22}$$

Assume that the shadowing and small-scale fading effects of different links between different users are independent. We can then generalize (22) as

$$L_{k,l} = \int_0^{\sqrt{2}} \left[\int_0^{\infty} \exp\left[-\frac{\sigma_n^2(2^C-1)}{E_{\mathrm{D}} s \mathrm{PG}(Dx)}\right] f_{s_{k,l}}(s)ds\right] f_{\mathrm{sq}}(x))dx, \tag{23}$$

where $f_{s_{k,l}}(s)$ is the pdf of the shadowing effect for the channel link between user k and l. Let the mean and standard deviation of the lognormal distribution be u_{dB} and σ_F, respectively. We then obtain (24) on the top of next page. It should be noted that the inner integral of (24) is the complement of the channel outage when the joint effect of the fading is the Suzuki distribution [44].

IV. Proposed Caching Policy Design

From the discussions in Sec. III.C, we understand that the proposed utility maximization can be used in solving many practical and important problems. We thus propose in this section a general solution approach for solving (7). Specifically, we propose an approach that iteratively optimizes the caching policies of users. Denote $\mathbf{b}_{k'} = [b_1^{k'}, ..., b_M^{k'}]^T$ as the policy vector of user k'. We iteratively solve the following subproblem for different k' by fixing other users' caching policies:

$$\max_{\mathbf{b}_{k'}} \quad U_{\mathrm{LP}}^{k'} = U_{\mathrm{net}}(\mathbf{b}_1, ..., \mathbf{b}_{k'}, ..., \mathbf{b}_K) \tag{25a}$$

$$\text{subject to} \quad \textstyle\sum_{m=1}^M b_m^{k'} = S, \tag{25b}$$

$$0 \le b_m \le 1, \forall m. \tag{25c}$$

When $k' \in \mathcal{U}_A$, we obtain the $U_{\mathrm{LP}}^{k'}$ expressed in (26) on the top of next page; when $k' \in \mathcal{U}_I$, we obtain the $U_{\mathrm{LP}}^{k'}$ expressed in (27) on the top of next page. Note that (26) and (27) are simply reformulations of (6), in which we isolate the terms that contain the variables to be optimized. From (26) and (27), we can see that (25) is a linear program.

General linear program solvers could be applied to solve (25). However, we provide here a more insightful and efficient approach via the analytical closed-form expressions in (26) and (27). By considering (28) on the top of next page, we notice that maximizing $U_{\mathrm{LP}}^{k'}$ is equivalent to maximizing

$$\sum_{m=1}^M b_m^{k'} U_{\mathrm{LP,S}}^{k',m}. \tag{29}$$

Then, observe that the optimal solution of (29), subject to constraints (25b) and (25c), can be obtained by allocating the cache space to the terms offering larger payoffs. Thus, the optimal solution of (25) is expressed as

$$(b_m^{k'})^* = \begin{cases} 1, & m \in \Phi_{k'}, \\ 0, & \text{otherwise}, \end{cases} \tag{30}$$

where

$$\Phi_{k'} = \{m : U_{\mathrm{LP,S}}^{k',m} \text{ is among the } S \text{ largest of all } U_{\mathrm{LP,S}}^{k',m}\}.$$

By iteratively solving (25) via (30) for different k' until convergence, the caching policy design problem in (7) can be effectively solved. Denote $\mathcal{B}_k = \{(b_1^k, ..., b_M^k)^T : \sum_{m=1}^M b_m^k = S; 0 \le b_m^k \le 1, \forall m\}$. The solution approach is summarized in Alg. 1. Since (30) suggests that the probability for a user to cache file m is either 1 or 0, we actually eliminate the probabilistic interpretation and attain the deterministic policies of users. To characterize the performance of the proposed solution approach, we provide the following theorem:

Theorem 1: Alg. 1 is monotonically non-decreasing at each iteration and can converge to a stationary point if each iteration provides a unique maximizer.[5]

Proof. See Appendix B. □

Algorithm 1 Iterative User-Based Caching Policy Design

```
At iteration $r$, choose a user $k'$ and update
$\mathbf{b}_{k'}^{r+1} = \arg\max_{\mathbf{b}_{k'} \in \mathcal{B}_{k'}} U(\mathbf{b}_1^r, ..., \mathbf{b}_{k'-1}^r, \mathbf{b}_{k'}, \mathbf{b}_{k'+1}^r, ..., \mathbf{b}_K^r)$
$\mathbf{b}_k^{r+1} = \mathbf{b}_k^r, \forall k \neq k'$
```

Finally, we note that although the proposed design needs coordination between users, the users can independently run

[5] If the maximizer is not unique, then we will encounter a tie between different $U_{\mathrm{LP,S}}^{k',m}$, which is generally unlikely, as users have different preferences on different files. Thus, such a unique maximizer assumption is mild.

$$L_{k,l} = \int_0^{\sqrt{2}} \left[\int_0^{\infty} \exp\left[-\frac{\sigma_n^2 (2^C - 1)}{E_{\mathrm{D}} \mathrm{PG}(Dx) s} \right] \frac{10/\log(10)}{s \sigma_F \sqrt{2\pi}} \exp\left(\frac{-(10 \log_{10}(s) - u_{dB})^2}{2\sigma_F^2} \right) ds \right] f_{\mathrm{sq}}(x) dx \tag{24}$$

$$\begin{aligned} U_{\mathrm{LP}}^{k'} = & \sum_{k \in \mathcal{U}_A} \frac{w_k U_{\mathrm{D}}}{K_{\mathrm{A}}} + (U_{\mathrm{B}} - U_{\mathrm{D}}) \sum_{m=1}^{M} \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} \left[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \right] + (K_{\mathrm{A}} U_{\mathrm{S}} - U_{\mathrm{D}}) \left[\sum_{m=1}^{M} \sum_{k \in \mathcal{U}_A, k \neq k'} \frac{w_k a^k_m b^k_m}{K_{\mathrm{A}}} \right] \\ & - \sum_{m=1}^{M} b^{k'}_m \left((U_{\mathrm{B}} - U_{\mathrm{D}}) \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} L_{k,k'} \left[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \right] + (U_{\mathrm{D}} - K_{\mathrm{A}} U_{\mathrm{S}}) \frac{w_{k'} a^{k'}_m}{K_{\mathrm{A}}} \right). \end{aligned} \tag{26}$$

$$\begin{aligned} U_{\mathrm{LP}}^{k'} = & \sum_{k \in \mathcal{U}_A} \frac{w_k U_{\mathrm{D}}}{K_{\mathrm{A}}} + (U_{\mathrm{B}} - U_{\mathrm{D}}) \sum_{m=1}^{M} \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} \left[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \right] + \\ & (K_{\mathrm{A}} U_{\mathrm{S}} - U_{\mathrm{D}}) \left[\sum_{m=1}^{M} \sum_{k \in \mathcal{U}_A} \frac{w_k a^k_m b^k_m}{K_{\mathrm{A}}} \right] - \sum_{m=1}^{M} b^{k'}_m \Bigg((U_{\mathrm{B}} - U_{\mathrm{D}}) \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} L_{k,k'} \Bigg[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \Bigg] \Bigg] \Bigg). \end{aligned} \tag{27}$$

$$U_{\mathrm{LP,S}}^{k',m} := \begin{cases} \left((U_{\mathrm{D}} - U_{\mathrm{B}}) \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} L_{k,k'} \left[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \right] + (K_{\mathrm{A}} U_{\mathrm{S}} - U_{\mathrm{D}}) \frac{w_{k'} a^{k'}_m}{K_{\mathrm{A}}} \right), & k' \in \mathcal{U}_A, \\ \Bigg((U_{\mathrm{D}} - U_{\mathrm{B}}) \sum_{k \in \mathcal{U}_A} \frac{w_k a_{m,k}}{K_{\mathrm{A}}} L_{k,k'} \Bigg[\prod_{l \in \mathcal{U}, l \neq k'} (1 - b^l_m L_{k,l}) \Bigg] \Bigg] \Bigg), & k' \in \mathcal{U}_I \end{cases} \tag{28}$$

the proposed Alg. 1 since they know the other users' individual preferences. In other words, Alg. 1 can actually be implemented in a decentralized manner, given that the users perfectly know the other users' individual preferences.

A. Complexity Analysis of the Proposed Caching Policy Design

From theorem 1, we can already understand the performance and convergence of the proposed design. Now, we analyze its complexity. Observe that the proposed design is based on the iterative algorithm in Alg. 1. Thus, the complexity comes from the computation at each iteration and the number of iterations required for convergence. At each iteration, the main computational complexity comes from computing for $U_{\mathrm{LP,S}}^{k,m}, \forall m$ and sorting $U_{\mathrm{LP,S}}^{k,m}, \forall m$. Then, we note that in terms of the total number of additions and multiplications, the complexity order when computing for $U_{\mathrm{LP,S}}^{k,m}, \forall m$ is $\mathcal{O}\left(MK^2\right)$; the complexity order of sorting is $\mathcal{O}(M \log M)$. As a result, the overall complexity order at each iteration becomes $\mathcal{O}\left(MK^2 + M \log M\right)$.

Regarding the number of iterations for the convergence, the general analytical expression is intractable; thus, we run simulations to understand how many iterations we would need in practice. For the simulations, we consider the same setup as that shown in Fig. 4 and evaluate the proposed throughput-based design (see Sec. V for the details of the simulation setup). The convergence of a single user's caching policy does not necessarily imply convergence of all users. Hence, we test the stopping criterion after updating the caching policies of all users in order to guarantee the convergence of all users. This is given as

$$\sum_{m=1}^{M} \sum_{k=1}^{K} |b^k_m|^2 \leq 10^{-4}. \tag{31}$$

Therefore, the number of iterations would be the multiple of K, i.e., the number of iterations can only be $K, 2K, 3K, ...$, etc. We consider three different numbers of users for the simulations: $K = 10, 20, 30$. The results show that the proposed design can converge within $10K$ number of iterations for more than 99% of the cases. In practice, this indicates fast convergence.

V. Numerical Results

This section provides simulation results to validate the analysis, evaluate the proposed designs, compare between different designs, and provide insights.

A. Simulation Setup

We evaluate here the performance of a cluster that covers a square area and has side length D. In the simulations, we assume that the users are uniformly distributed within the cluster. Unless otherwise indicated, we assume that the users adopt random-push scheduling. Likewise, we assume that users have equal weights, i.e., $w_k = 1, \forall k \in \mathcal{U}_A$. We consider a practical channel model for D2D links, which consists of the pathloss, shadowing, Rayleigh fading, and Gaussian noise.

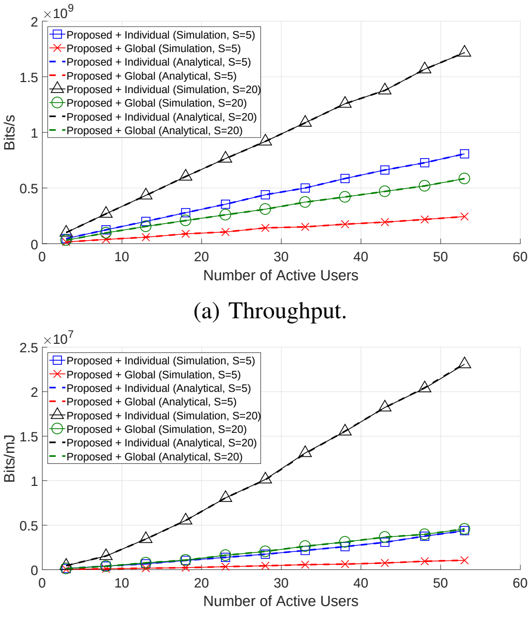
The path-loss model of the D2D link between users k and l is described as [8], [44]

$$20\log_{10}\frac{4\pi d_0}{\lambda_c} + 10\alpha\log_{10}\frac{d_{k,l}}{d_0}, \tag{32}$$

where $d_0 = 10$ m is the breakpoint distance, $\lambda_c = \frac{3\times10^8}{f_c}$ m, where $f_c = 2$ GHz is the carrier frequency, $\alpha = 3.68$ is the path-loss exponent, and $d_{k,l}$ is the distance between users k and l. The shadowing is modeled by a lognormal distribution with mean $\mu_{\text{dB}} = 0$ dB and standard deviation $\sigma_F = 8$ dB, and the small-scale fading is Rayleigh distributed. We assume that the noise power spectral density is $N_0 = -174$ dBm/Hz. We denote E_{D} as the transmission power of the device and $\text{SNR}_{\text{min}} = 5$ dB as the minimum SNR requirement for a successful transmission of a D2D link. Thus, $R_{\text{min}} = \log_2(1+3.16)$ is the minimum transmission rate of a D2D link. We then let $T_{\text{D}} = B_{\text{D}}R_{\text{min}}$ be the throughput of a D2D link, where $B_{\text{D}} = 20$ MHz is the bandwidth of a D2D link. We assume that a BS link will always exist whenever a user is scheduled to use it. Since the BS must supply the users in many clusters, we assume that a BS link can share only $\frac{1}{100}$ of the BS resources. Hence, the transmission power of a BS link is $E_{\text{B}} = 26$ dBm, which is $\frac{1}{100}$ of the total 46 dBm of the BS power. Similarly, the bandwidth of a BS link is $B_{\text{B}} = 200$ kHz, which is $\frac{1}{100}$ of the total 20 MHz bandwidth. We thus let $T_{\text{B}} = B_{\text{B}}R_{\text{min}}$. Note that a 200-kHz bandwidth is enough to transmit a low-resolution video, e.g., 360p. We assume that there is no cost when users obtain the desired file from their local caches, and we let $T_{\text{S}} = 2T_{\text{D}}$ to indicate the slightly better quality of the video when self-caching is possible.[6] For simplicity, we assume that the energy cost is purely determined by the radio frequency energy required for transmission; access to storage and coding/decoding is assumed to be negligible in comparison. Thus, based on the above setup, we obtain $C_{\text{B}} = E_{\text{B}}$, $C_{\text{D}} = E_{\text{D}}$, and $C_{\text{S}} = 0$. Therefore, the EE of the network is $\text{EE}_{\text{net}} = \frac{T_{\text{net}}}{C_{\text{net}}}$ according to the definition in Sec. III.C.5.

We consider $M = 1000$ for all simulations. To obtain the individual preferences of users and the corresponding system popularity for the simulations, we use the generator described in [33] to generate individual preference probabilities of 20000 users to form a dataset. Then, for each realization of the simulations, we randomly select users in the generated dataset for evaluation. As such, the system popularity of the simulations is simply the average of individual preferences of all 20000 users in the dataset.

In the following, we show the benefits of exploiting the individual preferences by comparing between designs with and without using individual preferences. In other words, we compare between the network performances where the proposed design is implemented either by using the knowledge of individual preference probabilities as in Sec. IV or simply by using the system-wide popularity distribution. In the latter case, the individual preference probabilities of all users in (14) are replaced by the global (system) popularity distribution, i.e., all users assume the same preference probabilities described by the global popularity distribution.



(a) Throughput.

(b) EE.

Fig. 2: Comparisons between analytical and simulated results in terms of throughput and EE.

B. *Effects of Individual Preferences*

In this subsection, we validate the analytical results provided in Sec. III and show the benefits of using information regarding individual preferences. For all simulations in this subsection, we adopt $D = 80$ m and $E_{\text{D}} = 20$ dBm. In the figures, the results of the proposed design that uses individual preferences is labeled with "+ Individual"; the design that uses global popularity distribution is labeled with "+ Global."

We first verify our analytical formulations and show the efficacy of using individual preferences. In Fig. 2, we consider both $S = 5$ and $S = 20$ and no inactive users ($K_{\text{I}} = 0$), and we evaluate the proposed design in terms of the throughput and EE. When evaluating the throughput in 2(a), the proposed design is used for maximizing the throughput; when evaluating the EE in 2(b), the proposed design is used for maximizing the EE. The curves labeled with "Analytical" are directly computed from expressions in Sec. III; the curves with "Simulations" are results of Monte Carlo simulations. We observe that the analytical results match the simulations very well, thereby validating our derivations in Sec. III. Moreover, we see that the design that exploits individual preferences significantly outperforms the corresponding design that does not use individual preferences.

We next show the impact of inactive users by observing the throughput difference. In Fig. 3, we consider $S = 5$ and compare the performances of two networks that have different numbers of inactive users, i.e., $K_{\text{I}} = 0$ and $K_{\text{I}} = 25$. The curves are generated by using the proposed design that aims to maximize throughput. The results show that the benefits

[6] Although we can immediately obtain the file when it is in the local cache, the throughput is bounded by the rate at which the user watches the file. Also, mathematically, we should not let T_{S} go to infinity if we want meaningful results.

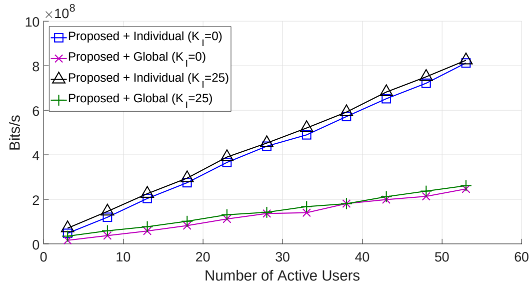


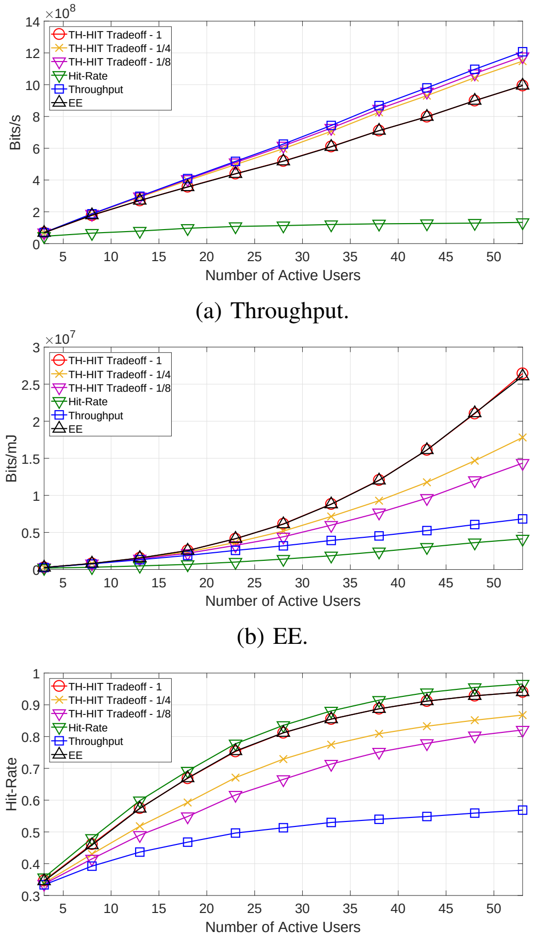
Fig. 3: Comparisons between networks with different numbers of inactive users in terms of throughput.

of the inactive users are more significant when the number of active users in a cluster is small. At $K_\text{A} = 3$, the throughput performance improves by 49% when $K_\text{I} = 25$; at $K_\text{A} = 53$, it improves by only 1.5% when $K_\text{I} = 25$. This indicates that although inactive users can help to improve performance, such improvement becomes insignificant when too many users (in the same cluster) share a single D2D band. This implies that when the number of inactive users is large, we might want to have multiple D2D links [48] to benefits more from the inactive users or adjust the number of users in a cluster by reducing the cluster size. However, the use of either approach should be subject to careful tradeoffs between different aspects, such as interference management, power control, reduction of hit-rate, etc.

C. *Tradeoff Behaviors between Different Performance Metrics*

In this subsection, we compare different designs and show the tradeoffs between throughput, EE, and hit-rate. Specifically, in all the following figures, we compare between caching policies obtained by using the proposed design framework in pursuit of different goals, i.e., throughput, EE, hit-rate, and the throughput–hit-rate tradeoff, in terms of throughput, EE, and hit-rate. For the throughput–hit-rate tradeoff design, we design the caching policies of users by maximizing $T_\text{net}+\zeta T_\text{D} K_\text{A} H_\text{net}$, i.e., by using $U_\text{B} = T_\text{B}$, $U_\text{D} = T_\text{D} + \zeta K_\text{A} T_\text{D}$, and $U_\text{S} = T_\text{S} + \zeta T_\text{D}$. Such a tradoff design is interpreted as a weighted sum of throughput and hit-rate in which the throughput is rendered the weight 1 and the hit-rate rendered the weight $\zeta K_\text{A} T_\text{D}$. Note that the term T_D in the weight of the hit-rate is basically to calibrate between different units. This tradeoff design is then labeled with “TH-HIT Tradeoff - ζ” in the figures, where ζ might be different to indicate different tradeoff behaviors.

Considering $S = 10$, $R = 80$, $E_\text{D} = 13$ dBm, and $K_\text{I} = 0$, we compare different designs in Fig. 4. Unsurprisingly, the throughput-based, EE-based, and hit-rate-based designs provide the best throughput, EE, and hit-rate, respectively. The hit-rate-based design provides poor throughput because it does not consider the self-caching gains possibly brought by letting users cache their desired files. In contrast, the throughput-based design is not effective in terms of hit-rate because the design overemphasizes self-caching gains. By striking a balanced viewpoint between throughput and hit-rate, the appropriate throughput–hit-rate tradeoff designs can efficiently trade throughput for hit-rate. This can then significantly improve the hit-rate without degrading the throughput much. Note that by adjusting ζ, we can effectively adjust the tradeoff behavior. Finally, we observe that an energy-efficient caching policy can be obtained by balancing the throughput and hit-rate. In fact, when $\zeta = 1$, the throughput–hit-rate tradeoff design performs almost as well as our proposed EE-based design. Also, it is worthwhile to note that when compared to the use of $E_\text{D} = 20$ dBm, as in Fig. 2, the adoption of $E_\text{D} = 13$ dBm here indeed reduces the power consumption significantly, resulting in much better EE. However, such a transmission power reduction only slightly increases the channel outage.[7] This implies the usefulness of a good power control policy of the network.



(a) Throughput.

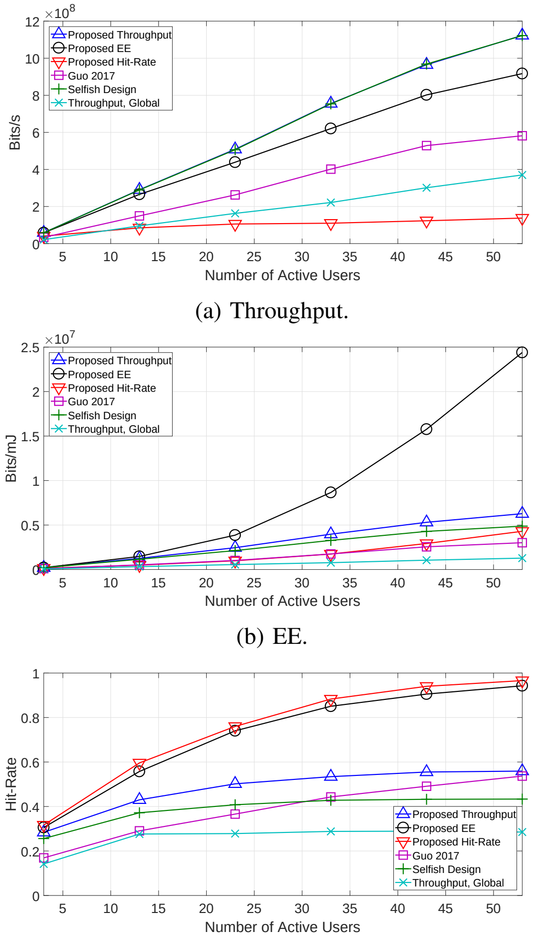
(b) EE.

(c) Hit-Rate.

Fig. 4: Comparisons between different designs in terms of throughput, EE, and hit-rate.

We now consider the same setup as featured in Fig. 4 and compare the proposed design to some other reference designs in Fig. 5. Specifically, we compare the proposed design to the baseline selfish design, in which each user selfishly caches the files according to its own preference without considering other users. The selfish design can be considered as an extreme as opposed to the maximum hit-rate

[7] The channel outage rate increases by only 0.012.



(a) Throughput.

(b) EE.

(c) Hit-Rate.

Fig. 5: Comparisons between different designs in terms of throughput, EE, and hit-rate.

design, which maximizes cooperation between users. We also compare the proposed designs to the design that adopts the global popularity distribution, similar to those in Figs. 2 and 3. Furthermore, we compare the proposed designs to Alg. 1 in [35], which is labeled as “Guo 2017.” To adapt the design in [35] to our network model, we make some revisions to it. First, we let each group defined in [35] stand for only a single user. Additionally, we let the cooperation range defined in [35] be the same as the cluster size defined in our paper. Since the design proposed in [35] assumes that a user can make caching decisions only for a single cache space, we implement a naive extension for it by treating each caching space of a user independently and repeatedly using the same policy designed in [35] for every caching space of the user.

The results show that our proposed designs outperform all the other designs in terms of throughput, EE, and hit-rate. Moreover, the selfish design performs well in terms of throughput, while it performs very poorly in terms of EE and hit-rate. This is because the network throughput can be effectively enhanced by having large local gains when all users are active. However, this near-optimal selfish design cannot hold if there are inactive users. We will see this later in another figure. On the other hand, the selfish design inherently provides very poor hit-rate, leading to poor EE, since the BS links are frequently used.

D. Performance Evaluations with Respect to Cluster Size

Fig. 6 and Fig. 7 evaluate the proposed designs with respect to the cluster size D. A change in the cluster size should be accompanied by a suitable transmission power control of D2D links. Hence, we adopt the power control policy proposed in [18] to appropriately manage the average SNR of the received signal and the interference between clusters. This power control policy is:

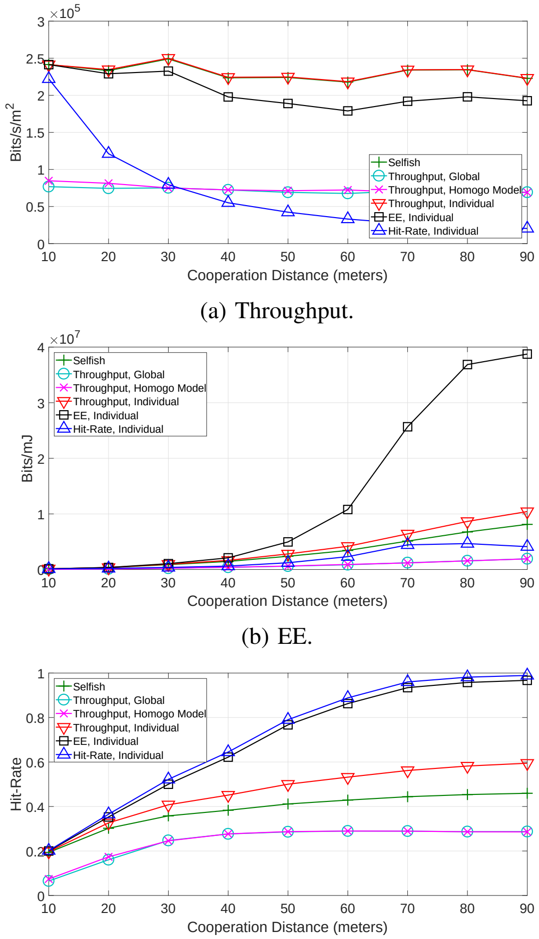
$$E_{\mathrm{D}} = \left[(\sqrt{K}-1)\frac{d}{d_0}\right]^{\alpha} \cdot (\frac{4\pi d_0}{\lambda_{\mathrm{c}}})^2 \cdot \nu, \tag{33}$$

where $K = 16$ is the reuse factor and $\nu = 2^{\frac{\alpha}{2}} N_0 B_{\mathrm{D}}$ is the maximum allowable interference between clusters.[8] Such a power control policy can adjust the transmission power of devices such that the average SNR of the received signal and the inter-cluster interference are almost invariant when changing the cluster size. Since D2D links are expected to exist only for short-distance transmissions, we consider $D \leq 90$ m. This results in $E_{\mathrm{D}} \leq 20$ dBm when we use (33) to adjust the power. We consider the Poisson point processes to model the number of active and inactive users in the cluster, where λ_{A} and λ_{I} represent the densities of active and inactive users, respectively. Thus, the numbers of active and inactive users are the random variables described by the Poisson distributions with parameter $\lambda_{\mathrm{A}} D^2$ and $\lambda_{\mathrm{I}} D^2$, respectively. Also, we need to accommodate the fact that a cluster has different numbers of users when D is different. Hence, instead of directly looking at the throughput, we evaluate using the throughput per area (Bits/s/m^2).

Similar to Fig. 4, we compare between different caching policies developed through the proposed design, with the aim of maximizing different performance metrics. In addition, we compare with two reference curves in Fig. 5. Furthermore, we include an additional reference curve, which adopts the coordinated design with homogeneous modeling (labeled “Homogo Model” in Figs. 6 and 7). This curve considers the situation that the policy is designed using global popularity distribution while the users indeed have the same preference, following the global popularity distribution. It thus represents the performance of systems that design and evaluate using the homogeneous modeling employed in previous papers – we want to see the influences on the performance evaluation of cache-aided D2D networks when changing from a homogeneous modeling to a more practical heterogeneous modeling.

In Fig. 6, we consider $S = 10$, $\lambda_{\mathrm{A}} = 0.01$, and $\lambda_{\mathrm{I}} = 0$, i.e., no inactive users. Since the area throughput and EE are influenced by multiple factors, they are not convex/concave functions in general. As such, we see that the area throughput of the throughput-based design fluctuates when D is small; it becomes somewhat flat when D is large. This is because the contribution of the D2D transmission becomes minor

[8] The value of ν is at the level of noise power; hence, for brevity, we ignore the inter-cluster interference in the simulations. Accordingly, we use ν here to compute only for E_{D}.

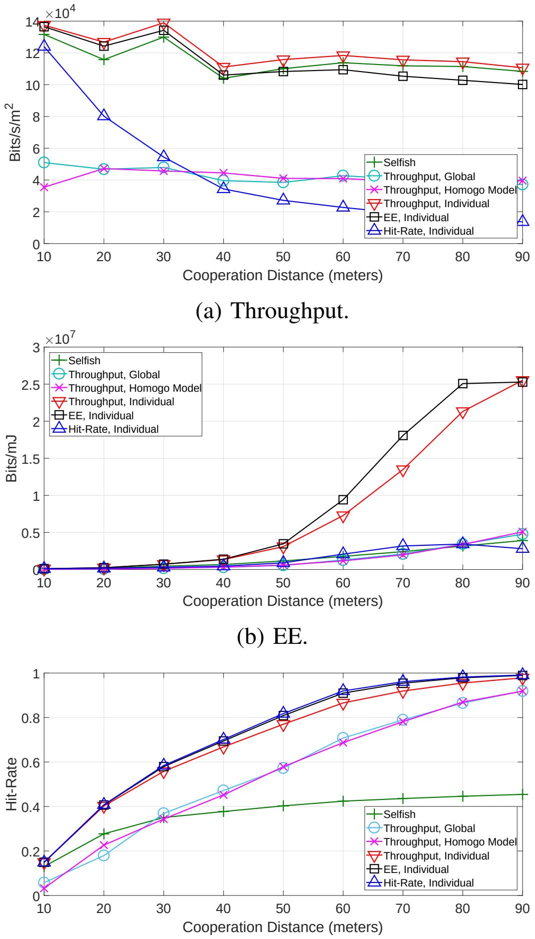


(a) Throughput.

(b) EE.

(c) Hit-Rate.

Fig. 6: Comparisons between different designs in terms of throughput, EE, and hit-rate with respect to cluster size with $\lambda_{\mathrm{A}} = 0.01$ and $\lambda_{\mathrm{I}} = 0$.



(a) Throughput.

(b) EE.

(c) Hit-Rate.

Fig. 7: Comparisons between different designs in terms of throughput, EE, and hit-rate with respect to cluster size with $\lambda_{\mathrm{A}} = 0.005$ and $\lambda_{\mathrm{I}} = 0.005$.

as too many users share the same D2D band in a cluster. We also see that the selfish design is relatively effective again because all users are active. As expected, the hit-rate-based design provides the best hit-rate. Meanwhile, the area throughput of the hit-rate-based design continuously decreases with respect to D, since it strives to improve the hit-rate without considering the influence of the self-caching gain. In contrast, the throughput-based design again provides the best area throughput, but it is not effective for hit-rate. In terms of EE, the EE-based design outperforms others significantly. Moreover, the optimal point of the EE is at a large cooperation distance because it is necessary to have high hit-rate in order to have large EE; otherwise, the BS needs to serve the users by using more BS links, leading to the smaller throughput and larger power consumption, and thus poor EE.

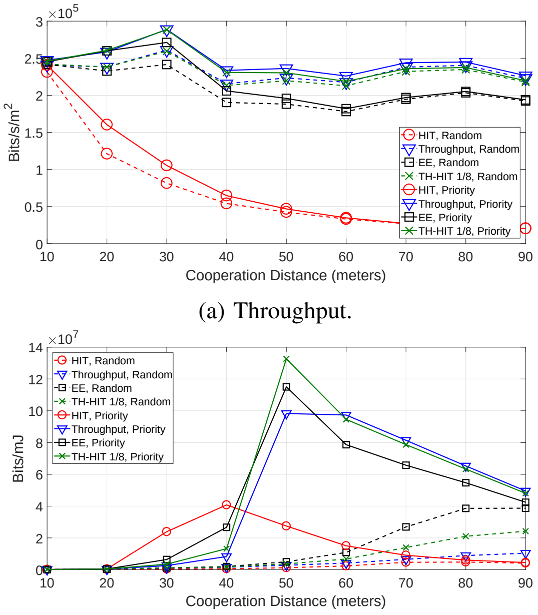
Through observation, we see that the tradeoff between the area throughput and EE can be attained not only through different caching policies but also through different cluster sizes. Thus, a network designer should consider both the caching policy and cluster size when designing the network. Finally, we argue that exploiting individual preferences is expectedly beneficial. We can see that our proposed system, which considers individual preferences in the design, performs better than the system that operates under the assumption that users have the same preferences, i.e., the curve with the “Homogo Model” label. Such a result implies that, rather than being detrimental, the diverse preferences of users for files can actually be used to further improve the network.

In Fig. 7, we conduct a similar evaluation as in Fig. 6. Here, we adopt $\lambda_{\mathrm{A}} = 0.005$ and $\lambda_{\mathrm{I}} = 0.005$, i.e., there are some inactive users. We can see that most of the phenomena observed in Fig. 6 can be observed again here. Since we now have inactive users, they should be cooperative such that we can obtain the optimal throughput while the active users are still fairly selfish. This then distinguishes the selfish design from our proposed design and causes the throughput-based design to perform well in terms of EE and hit-rate. We can actually observe that the difference between the optimal values of the throughput- and EE-based designs in terms of EE is smaller as compared to that in Fig. 6. However, the tradeoff between throughput and EE is still significant as we change the cluster size. Finally, we see that the proposed system performs better than the system with pure homogeneous modeling. This again validates our point that having users with diverse preferences is beneficial.

E. Performance Evaluations with Different Schedulers

Finally, we evaluate the proposed design in clustering networks with two different schedulers. Through this, we can show how the proposed design can help developers design a caching policy for a network that has a very complicated scheduler. Specifically, in addition to evaluating using the random-push scheduler, we evaluate (under the same caching policy) the "priority-push scheduler" [18], which functions as follows: all users initially check whether the files in their local caches can satisfy their requests. If yes, then the requests are satisfied; otherwise, they send their requests to the BS. The BS then checks whether there are users that can be satisfied by using D2D links. If yes, then the BS randomly selects one user to be served by the D2D link; otherwise, the BS randomly selects one user from those sending the requests and then serves the user via a BS link. Such a scheduler maximizes the usage of D2D communications. Hence, we can expect that the priority-push network will have higher throughput and better EE than the random-push network. On the other hand, it might be unfair to those users whose preferences are not similar to the mainstream; they might be less likely to be selected and accordingly be served. More importantly, such a complicated scheduler results in an intractable expression for designing caching policies. We demonstrate how to exploit the proposed designs in this work along with some numerical results to guide the designer in obtaining effective designs for it.

In Fig. 8, we consider the same setup as in Fig. 6 and evaluate the proposed design in both networks using the random-push and priority-push schedulers, labeled as "Random" (dashed line) and "Priority" (solid line), respectively. We observe that the priority-push network generally outperforms the random-push network in terms of the area throughput and EE.[9] Additionally, we observe that in terms of the area throughput, the results of the random-push network can be fairly representative. The results for EE show more subtle effects. We see that the optimal cluster size for the priority-push network is much smaller, which implies that it is unnecessary in the priority-push network to have high hit-rate to obtain the best EE. This is because the priority-push scheduler would schedule a D2D link as long as there exists one, implying that it would have a higher rate for scheduling D2D links than simply the hit-rate – the probability for at least one user to find the desired file in the D2D network is higher than that of a particular user to find its desired file. Thus, to obtain an energy-efficient design in the priority-push network, we need to choose a design that has smaller cluster size and has lower hit-rate than those that provide the optimal EE in the random-push network. Overall, based on the above results, we conclude that we need to reduce the cluster size and consider various tradeoff designs proposed in this paper in order to obtain an effective design in the priority-push network. Since our proposed tradeoff designs can efficiently evaluate the throughput and hit-rate, such a trial-and-error procedure might not be challenging.



(a) Throughput.

(b) EE.

Fig. 8: Comparisons between different designs in terms of throughput and EE with respect to cluster size with $\lambda_{\mathrm{A}} = 0.01$ and $\lambda_{\mathrm{I}} = 0$.

F. Summary of the Insights

Here we summarize the insights from our simulation results:

- It is necessary to consider the influence of users' individual preferences on system design because evaluations done under the assumption that users have the same preferences are not representative of evaluations done while considering individual preferences. Therefore, by considering the effects of individual preferences, the proposed designs can significantly improve network performance.
- A system optimized for throughput can have significant loss in EE and vice versa. Similarly, a system optimized for hit-rate can have significant loss in throughput and EE and vice versa. However, allowing slight sub-optimality in one performance can significantly improve another performance.
- Instead of directly optimizing EE, which can be complicated, one alternative approach is to use a throughput–hit-rate tradeoff design, i.e., by balancing between throughput and hit-rate, a design with high EE can be obtained.
- While directly finding the effective caching policy for the network using priority-push scheduling is very challenging due to mathematical intractability, various throughput–hit-rate tradeoff designs obtained by considering tractable random-push scheduling can serve as alternatives. This is because priority-push and random-push scheduling have similar behaviors, except that the former scheduling prioritizes the users that can use D2D links.

[9] Since the hit-rate considering the priority-push scheduling is the same as the hit-rate considering the random-push scheduling, we omit the demonstration of the hit-rate for brevity.

VI. Conclusions

In this work, we used the individual preferences of users to improve cache-aided D2D networks. We used an individual preference probability model to derive the network utility of a clustering network and to propose a utility maximization problem accordingly. This problem can be applied to solve different important and practical problems, e.g., throughput, EE, hit-rate optimization, and different tradeoff problems. Assuming users can coordinate, we proposed a general solution approach for solving the utility maximization problem. Comprehensive numerical evaluations were conducted with practical individual preferences and network setups. Our results show that we can appropriately exploit information about users' individual preferences to significantly increase the performance of cache-aided D2D networks. Our results also show that throughput and hit-rate are in conflict with each other; nevertheless, such a conflict can be resolved through a suitable tradeoff design. To obtain an effective EE design, in addition to directly optimizing EE, we can solve a properly designed throughput–hit-rate tradeoff design, offering another perspective for EE optimization. Aside from optimizing the caching policy to improve performance, we proposed changing the cooperation distance of the D2D to achieve this goal; likewise, the tradeoff exists in this regard. Finally, we demonstrated that the results of our work can serve as a foundation for designing caching policies in networks with a more involved scheduling policy.

Appendix A
Derivations of the Expected Utility

We first derive the expression of U. Using (1), (2), and (3), we obtain (34) on the top of next page. It follows that

$$
\begin{aligned}
&U_{\text{net}} = \\
&\sum_{k\in\mathcal{U}_A}\frac{w_kU_{\text{D}}}{K_{\text{A}}} + (U_{\text{B}}-U_{\text{D}})\sum_{m=1}^{M}S_m + \\
&(U_{\text{S}}-U_{\text{D}})\sum_{m=1}^{M}\sum_{k\in\mathcal{U}_A}\frac{w_ka_m^kb_m^k}{K_{\text{A}}} + U_{\text{S}}\sum_{m=1}^{M}\sum_{k\in\mathcal{U}_A}\sum_{l\in\mathcal{U}_A,l\neq k}\frac{w_la_m^lb_m^l}{K_{\text{A}}}\\
&=\sum_{k\in\mathcal{U}_A}\frac{w_kU_{\text{D}}}{K_{\text{A}}} + (U_{\text{B}}-U_{\text{D}})\sum_{m=1}^{M}S_m\\
&\qquad + (K_{\text{A}}U_{\text{S}}-U_{\text{D}})\sum_{m=1}^{M}\sum_{k\in\mathcal{U}_A}\frac{w_ka_m^kb_m^k}{K_{\text{A}}}.
\end{aligned}
$$

Appendix B
Proof of Theorem 1

To prove the Theorem, we first note that the problem in (7) satisfies the block separable structure as follows:

$$
\begin{aligned}
\max_{\mathbf{b}_1,...,\mathbf{b}_K}\quad & U(\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_K)\\
\text{s.t.}\quad & \mathbf{b}_k\in\mathcal{B}_k,\forall k.
\end{aligned}
\tag{35}
$$

Eq. (35) indicates that the constraints on different blocks are separable. Denote $u(\mathbf{b}_{k'};\mathbf{B}^r) = U(\mathbf{b}_1^r,...,\mathbf{b}_{k'-1}^r,\mathbf{b}_{k'},\mathbf{b}_{k'+1}^r,...,\mathbf{b}_K^r)$ for brevity. From Alg. 1, we notice that $\mathbf{b}_{k'}^{r+1} = \arg\max_{\mathbf{b}_{k'}\in\mathcal{B}_i} u(\mathbf{b}_{k'};\mathbf{B}^r)$ at each iteration. Hence, we have

$$
u(\mathbf{b}_{k'}^{r+1};\mathbf{B}^r) \geq u(\mathbf{b}_{k'}^{r};\mathbf{B}^r). \tag{36}
$$

Thus, we know that the algorithm is monotonically non-decreasing. Then, since the optimal objective function of (7) should not be infinity, the algorithm must converge.

To prove that Alg. 1 converges to a stationary point if every iteration has an unique maximization, we use the analysis framework for block coordinate descent methods in [47] as follows.[10] Suppose each iteration in Alg. 1 has a unique maximizer. Then, Alg. 1 converges to a unique solution $\bar{\mathbf{B}} = \left[\bar{\mathbf{b}}_1,...,\bar{\mathbf{b}}_K\right]$ as the number of iterations $r\to\infty$. According to Alg. 1, we know that

$$
\bar{\mathbf{b}}_{k'} = \arg\max_{\mathbf{b}_{k'}\in\mathcal{B}_{k'}} u(\mathbf{b}_{k'};\bar{\mathbf{B}}),\forall k'=1,2,...,K. \tag{37}
$$

As a result, due to concavity,

$$
\left(\nabla_{k'}U(\bar{\mathbf{b}}_1,...,\bar{\mathbf{b}}_K)\right)^T\left(\mathbf{b}_{k'}-\bar{\mathbf{b}}_{k'}\right)\leq 0,\forall\mathbf{b}_{k'}\in\mathcal{B}_{k'}, \tag{38}
$$

where $\nabla_{k'}$ denotes the gradient of U with respect to component $\mathbf{b}_{k'}$. We denote $\mathbf{b} = vec(\mathbf{B})\in\mathcal{B}$ is the vectorization of $\mathbf{B} = [\mathbf{b}_1,..,\mathbf{b}_K]$ and $\bar{\mathbf{b}} = vec(\bar{\mathbf{B}})\in\mathcal{B}$ is the vectorization of $\bar{\mathbf{B}}$, where $\mathcal{B} = vec(\mathcal{B}_1\times\mathcal{B}_2\times...\times\mathcal{B}_K)$. Then, notice that (38) is true for all $k'=1,2,...,K$. It follows from the Cartesian product structure of a set that

$$
\left(\nabla U(\bar{\mathbf{b}})\right)^T\left(\mathbf{b}-\bar{\mathbf{b}}\right)\leq 0,\forall\mathbf{b}\in\mathcal{B}, \tag{39}
$$

where ∇ is the gradient with respect to $\bar{\mathbf{b}}$. This proves that Alg. 1 converges to a stationary point.

References

[1] M.-C. Lee and A. F. Molisch, "Individual preference aware caching policy design for energy-efficient wireless D2D communications," *in Proc. IEEE GLOBECOM*, Dec. 2017.

[2] "Cisco Virtual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021," San Jose, CA, USA.

[3] J. G. Andrews, S. Buzzi, W. Choi, *et. al.*, "What will 5G be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065-1082, Jun. 2014.

[4] A. F. Molisch, G. Caire, D. Ott, J. R. Foerster, D. Bethanabhotla, and M. Ji, "Caching eliminates the wireless bottleneck in video aware wireless networks," *Adv. Elect. Eng.,* vol. 2014, Nov. 2014, Art. ID 261390.

[5] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commmun. Mag.*, vol. 54, no. 9, pp. 22-28, Sep. 2016.

[6] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commmun. Mag.*, vol. 51, no. 4, pp. 142-149, Apr. 2013.

[7] 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.

[8] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Area Commun.*, vol. 34, no. 1, pp. 176-189, Jan. 2016.

[9] M. Ji, G. Gaire, and A. F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833-6859, Dec. 2015.

[10] M. Ji, G. Caire, and A. F. Molisch "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849-864, Feb. 2016.

[10] The proof basically follows the same steps as those taken to prove Proposition 2.7.1 in [47]. Hence, we here provide a shortened version of the proof.

$$\begin{aligned} U &= \sum_{k\in\mathcal{U}_A} \frac{w_k}{K_\mathrm{A}} \mathbb{E}\Bigg\{ U_\mathrm{D} \left(1 - \sum_{m=1}^{M} a_m^k \left[\prod_{l\in\mathcal{U}} (1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}) \right] - \sum_{m=1}^{M} a_m^k b_m^k \right) \\ &\quad + U_\mathrm{B} \sum_{m=1}^{M} a_m^k \left[\prod_{l\in\mathcal{U}} (1 - b_m^l \mathbf{1}_{\{h_{k,l},C\}}) \right] + U_\mathrm{S} \sum_{m=1}^{M} a_m^k b_m^k \Bigg\} \\ &= \sum_{k\in\mathcal{U}_A} \frac{w_k U_\mathrm{D}}{K_\mathrm{A}} + (U_\mathrm{B} - U_\mathrm{D}) \sum_{k\in\mathcal{U}_A} \sum_{m=1}^{M} \frac{w_k a_m^k}{K_\mathrm{A}} \left[\prod_{l\in\mathcal{U}} (1 - b_m^l E_h\{\mathbf{1}_{\{h_{k,l},C\}}\}) \right] + (U_\mathrm{S} - U_\mathrm{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{w_k a_m^k b_m^k}{K_\mathrm{A}} \\ &= \sum_{k\in\mathcal{U}_A} \frac{w_k U_\mathrm{D}}{K_\mathrm{A}} + (U_\mathrm{B} - U_\mathrm{D}) \sum_{m=1}^{M} \underbrace{\sum_{k\in\mathcal{U}_A} \frac{w_k a_m^k}{K_\mathrm{A}} \left[\prod_{l\in\mathcal{U}} (1 - b_m^l L_{k,l}) \right]}_{S_m} + (U_\mathrm{S} - U_\mathrm{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{w_k a_m^k b_m^k}{K_\mathrm{A}} \\ &= \sum_{k\in\mathcal{U}_A} \frac{w_k U_\mathrm{D}}{K_\mathrm{A}} + (U_\mathrm{B} - U_\mathrm{D}) \sum_{m=1}^{M} S_m + (U_\mathrm{S} - U_\mathrm{D}) \sum_{m=1}^{M} \sum_{k\in\mathcal{U}_A} \frac{w_k a_m^k b_m^k}{K_\mathrm{A}}. \end{aligned} \tag{34}$$

[11] M.-C. Lee, M. Ji, A. F. Molisch, and N. Sastry, "Throughput-outage analysis and evaluation of cache-aided D2D networks with measured popularity distributions," *IEEE Trans. Wireless Commun.,* vol. 18, no. 11, pp. 5316-5332, Nov. 2019.

[12] K. Shanmugam, N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[13] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," *IEEE ICC*, Jun. 2015.

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

[15] U. Niessen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146-1158, Feb. 2017.

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

[17] B. Chen, C. Yang, and G. Wang, "High-throughput opportunistic cooperative device-to-device communications with caching," *IEEE Trans. Veh. Technol.,* vol. 66, no. 8, pp. 7527-7539, Aug. 2017.

[18] M.-C. Lee and A. F. Molisch, "Caching policy and cooperation distance design for base station assisted wireless D2D caching networks: Throughput and energy efficiency optimization and tradeoff," *IEEE Trans. Wireless Commun.,* vol. 17, no. 11, pp. 7500-7514, Nov. 2018.

[19] B. Chen, C. Yang, and A. F. Molisch, "Cache-enabled device-to-device communications: Offloading gain and energy cost," *IEEE Trans. Wireless Commun.,* vol. 17, no. 7, pp. 4519-4536, Jul. 2017.

[20] 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 Commun.,* vol. 65, no. 10, pp. 4365-4380, Oct. 2016.

[21] M. Naslcheraghi, M. Afshang, and H. S. Dhillon, "Modeling and performance analysis of full-duplex communications in cache-enabled D2D networks," *in Proc. IEEE ICC,* May 2018.

[22] J. Guo, J. Yuan, and J. Zhang, "An achievable throughput scaling law of wireless device-to-device caching networks with distributed MIMO and hierarchical Cooperations," *IEEE Trans. Wireless Commun.,* vol. 17, no. 1, pp. 492-505, Jan. 2018.

[23] A. Liu, V. K. N. Lau, and G. Caire, "Cache-induced hierarchical cooperation in wireless device-to-device caching networks," *IEEE Trans. Inf. Theory,* vol. 62, no. 6, pp. 4629-4652, Jun. 2018.

[24] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.,* vol. 16, no. 8, pp. 5001-5015, Aug. 2017.

[25] N. Giatsoglou, K. Ntontin, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "D2D-aware device caching in mmWave-cellular networks," *IEEE J. Sele. Area Commun.,* vol. 35, no. 9, pp. 2025-2037, Sep. 2017.

[26] M.-C. Lee, H. Feng, and A. F. Molisch, "Dynamic caching content replacement in base station assisted wireless D2D caching networks," *IEEE Access*, vol. 8, pp. 33909-33925, Feb. 2020.

[27] 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 Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 1710-1732, 2018.

[28] I. Ahmed, M. H. Ismail, and M. S. Hassan, "Video transmission using device-to-device communications: A survey," *IEEE Access,* vol. 7, pp. 131019-131038, 2019.

[29] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, "Device-enhanced mec: Multi-access edge computing (mec) aided by end device computation and caching: A survey," *IEEE Access,* vol. 7, pp. 166079-166108, 2019.

[30] D. Prerna, R. Tekchandani, and N. Kumar, "Device-to-device content caching techniques in 5G: A taxonomy, solutions, and challenges," *Elsevier Computer Communications,* vol. 153, no. 1, pp. 48-84, Mar. 2020.

[31] D. Karamshuk, N. Sastry, M. Al-Bassam, A. Secker, and J. Chandaria, "Take-away TV: Recharging wok commutes with predictive preloading of catch-up TV content," *IEEE J. Sel Commun.*, vol. 34, no. 8, pp. 2091-2101, Aug. 2016.

[32] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling for video content in wireless caching networks," *in Proc. IEEE GLOBECOM*, Dec. 2017.

[33] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling and parameterization for video content in wireless caching networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 676-690, Apr. 2019.

[34] D. Liu and C. Yang, "Caching at base stations with heterogeneous user demands and spatial locality," *IEEE Trans. Commun.,* vol. 67, no. 2, pp. 1554-1569, Feb. 2019.

[35] Y. Guo, L. Duan, and R. Zhang, "Cooperative local caching under heterogeneous file preferences," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 444-457, Jan. 2017.

[36] Y. Pan, C. Pan, H. Zhu, *et. al.,* "On consideration of content preference and sharing willingness in D2D assisted offloading," *IEEE J. Sel Commun.*, vol. 35, no. 4, pp. 978-993, Apr. 2017.

[37] B. Chen and C. Yang, "Caching policy optimization for D2D communications by learning user preference," *in Proc. IEEE VTC-Spring*, Jun. 2017.

[38] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for device-to-device caching networks," *IEEE System J.,* Dec. 2017.

[39] Y. Li, C. Zhong, M. C. Gursory, and S. Velipasalar, "Learning-based delay-aware caching in wireless D2D caching networks," *IEEE Access,* vol. 6, pp. 77250-77264, Nov. 2018.

[40] Y. Zhang, Y. Yu, Q. Wu, *et. al.,* "A game-theoretic approach for optimal distributed cooperative hybrid caching in D2D networks," *IEEE Wireless Commun. Lett.,* vol. 7, no. 3, pp. 324-327, Jun. 2018.

[41] L. Zhang, M. Xiao, G. Wu, and S. Li, "Efficient scheduling and power allocation for D2D-assisted wireless caching networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2438-2452, June 2016.

[42] M. Ehrgott, *Multicriteria Optimization.* Springer-Verlag New York, 2005.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[44] A. F. Molisch, *Wireless Communications*, IEEE Press-Wiley, 2nd ed., 2012.

[45] A. M. Mathai, *An Introduction to Geometrical Probability: Distributional Aspects with Applications*. Boca Raton, FL, USA: CRC, 1999.

[46] C.-C. Tseng, H.-T. Chen, and K.-C. Chen, “On the distance distributions of the wireless ad hoc networks,” *in Proc. IEEE VTC-Spring*, 2006.

[47] D. P. Bertsekas, *Nonlinear Programming,* Belmont: Athena scientific, 1999.

[48] Y. Cai and A. F. Molisch, “On the multi-activation oriented design of D2D-aided caching networks,” *in Proc. IEEE GLOBECOM*, Dec. 2019.