This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal

# Joint Caching and Routing in Congestible Networks of Arbitrary Topology

Boxi Liu, *Student Member, IEEE,* Konstantinos Poularakis, *Member, IEEE,* Leandros Tassiulas, *Fellow, IEEE* and Tao Jiang, *Fellow, IEEE*

***Abstract*—In-network caching constitutes a promising approach to reduce traffic loads and alleviate congestion in both wired and wireless networks. In this paper, we study the joint caching and routing problem in congestible networks of arbitrary topology (JoCRAT) as a generalization of previous efforts in this particular field. We show that JoCRAT extends many previous problems in the caching literature that are intractable even with specific topologies and/or assumed unlimited bandwidth of communications. To handle this significant but challenging problem, we develop a novel approximation algorithm with guaranteed performance bound based on a randomized rounding technique. Evaluation results demonstrate that our proposed algorithm achieves near-optimal performance over a broad array of synthetic and real networks, while significantly outperforming the state-of-the-art methods.**

***Index Terms*—Content caching, request routing, joint optimization, approximation algorithms, network congestion.**

## I. Introduction

### A. Motivation

In-network caching promises to benefit not only classic inter-networking architectures, e.g., Content Delivery Networks (CDN) [1], but also the emerging fifth generation (5G) and Internet of Things (IoT) systems [2]. For instance, caching data in intermediate nodes facilitates IoT end-users to postpone communication when the batteries are low and then recover it once having harvested enough energy [3]. Besides, in-network caching reduces the latency of retrieving both transient (e.g., sensing data [4]) and non-transient (e.g., video clips [5]) content from remote cloud centers for end-users [6].

Compared to other networked systems, IoT is expected to have at least tenfold increase in the number of end-users, which however merely rely on the bandwidth-limited wireless backhaul links among the IoT network devices [7]. To handle this major connectivity issue, a key challenge in IoT-oriented caching is to optimally place contents in caches distributed over a geographic area (*caching policy*) as well as to efficiently route content requests of end-users to them (*routing policy*).

Most of the existing works that investigate caching have assumed that network bandwidth is *abundant*, and hence can be freely leveraged to route requests from users to caches and fetch contents from them instead of remote cloud centers (e.g., see [8], [9], [10], [11]). This assumption has greatly simplified the caching problem. However, such simplification may be problematic in emerging 5G and IoT scenarios where the link bandwidth capacities are at risk of being overwhelmed (*congested*) by the massive traffic volume generated by the IoT end-users. Motivated by such scenarios, in this paper, we argue that caching needs to be *jointly* designed with the routing policy so as to balance the traffic load at the different links in the network and effectively increase the number of requests served by the caches.

Recently, the interplay between caching and routing policies has attracted the interest of the research community. Various *joint caching and routing (JCR)* works have focused on the theoretical analysis and approximation algorithms of this problem (e.g., see [12], [13], [14], [15], [16] and the discussion on related works in Section II). However, most of these works concentrated on *some specific network topologies* originated from classic networked systems, such as hierarchical (e.g., IPTV in CDN [12]) or 2-tier (e.g., macro/femto cell [13], [14], [15], [16]) networks where the available routing options are limited (e.g., a single routing option from the bottom to the top layers in the hierarchical network or the single-hop routing options in the 2-tier network). Clearly, this is not a practical assumption in IoT architectures where caches will be largely deployed at the edge of the network (e.g., at macro-cell base stations (BSs), femto-cells, edge cloud servers, fog nodes and gateways) and multiple routing paths will often exist between end-users and cache-nodes, possibly of multiple hops.

Practically, network operators are urging to leverage *the routing diversity* to maximize the benefit of caching and computation resources, e.g., to enable as many end-users access the in-network caches as possible through multi-hop multi-path connections [17], [18]. But nonetheless, departing from the previous simplified topologies, jointly optimizing the multi-hop multi-path connections together with content placement introduces a huge number of inter-

This work was supported in part by the National Science Foundation of China with Grant number 61729101, Major Program of National Natural Science Foundation of Hubei in China with Grant 2016CFA009, Fundamental Research Funds for the Central Universities with Grant number 2015ZDTD012, and the National Science Foundation of USA with Grant CNS 1815676. K. Poularakis acknowledges the Bodossaki Foundation, Greece, for a postdoctoral fellowship.

B. Liu and T. Jiang are with Wuhan National Laboratory for Optoelectronics and the School of Electronics Information and Communications, Huazhong University of Science and Technology, China (tao.jiang@ieee.org). K. Poularakis and L. Tassiulas are with the Department of Electrical Engineering and Institute for Network Science, Yale University, USA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



coupling variables into the JCR problem (see the analysis in Section III). To the best of our knowledge, few studies have addressed the theoretical aspects of this particularly important JCR variant. To embrace the in-network caching for IoT vision, comprehensive understanding of the JCR is required to properly manage the limited network resources allocated to end-users.

### B. Methodology and contributions

Motivated by the above discussion, in this paper, we study the *joint content caching and request routing problem in congestible networks of arbitrary topology (JoCRAT)*, with the objective of maximizing the volume of requests served by the caches. This objective is critical in scenarios of the cache-capable IoT vision where the capacity of the network may not suffice to serve all the content requests (e.g., see the works in [12], [13], [19] that used the same objective). We formulate the JoCRAT problem as an integer linear program. We analyze the complexity of this problem and show that for the special case of one cache the objective function admits the property of submodularity [20]. The submodularity property is theoretically attractive since there are several greedy algorithms with the tightest approximation ratio known for this class of problems. Interestingly, we find a counter-example showing that this attractive property does not hold for the general case of our problem and therefore greedy algorithms perhaps are no longer suitable.

To solve the problem for the general case, we use a randomized rounding technique along with an evaluation of the risk of overwhelmed capacities approach and develop a novel JCR algorithm with approximation guarantees. The approximation ratio is sub-linear to the network size and the minimum of cache and bandwidth capacities. The development of such an approximation algorithm is of value both in suggesting robust heuristic approaches to the intractable JCR, and in providing a further understanding of the elusive structure of their optima. Evaluation results on both synthetic and real network topologies reveal that our proposed algorithm achieves close-to-optimal performance, while significantly outperforming state-of-the-art caching schemes.

The technical contributions of this work can be summarized as follows:

- *JoCRAT Problem*. We model the problem of jointly designing the content caching and request routing policies in congestible networks of arbitrary topology (JoCRAT). This problem, primarily motivated by the emerging scenarios in cache-capable IoT vision, extends many previous works that consider specific network topologies and/or assumed non-overlapped routing paths as well as abundant bandwidth of the paths connecting caches and end-users.
- *Complexity Analysis*. We show that the JoCRAT problem is NP-Hard and does not have the property of submodularity which is commonly used in the caching literature to derive approximation algorithms. We manage to show this property for the special case of one cache only.
- *Approximation Algorithms*. We derive an algorithm with approximation guarantees for the JoCRAT problem by using a randomized rounding technique, appropriately tailored to our problem. To the best of our knowledge, this is the first approximation algorithm for the JCR problem in its general form.
- *Evaluation Results*. We conduct evaluations for various network topologies, both real and synthetic, and file request patterns. We show that, in practice, the proposed algorithm performs close-to-optimal and much better than the worst-case approximation guarantees suggest. Compared with four state-of-the-art caching schemes, our approach significantly increases the volume of requests served by the caches, especially for low and moderate bandwidth capacity scenarios.

The rest of the paper is organized as follows. Section II gives a comprehensive overview of related works while section III presents the system model and the formulation of the JoCRAT problem. We analyze the complexity of this problem and derive approximation algorithms in Sections IV and V, respectively. The evaluation results are presented in Section VI. We conclude our work in Section VII.

## II. Related Works

Broadly speaking, the algorithms for caching content are classified into *reactive* and *proactive*. Reactive caching is a popular technique that places content in caches on-demand. Examples include the Least Frequently Used (LFU) and Least Recently Used (LRU) algorithms as well as more advanced schemes based on machine learning [21]. On the other hand, proactive caching first estimates content demand patterns for some time period of interest (e.g., a few hours) and then places content in caches to meet the demand efficiently. Proactive caching algorithms are simple to implement and have been proven to improve performance over reactive caching when the demand can be estimated accurately [22]. Hence, this work is focused on proactive caching.

Proactive caching is a well-investigated problem in the literature (e.g., see [23] for a recent survey). This problem is NP-Hard in general due to its *combinatorial nature*, i.e., a binary decision for the placement of each content to each cache is needed. Therefore, previous works have focused on designing approximation and heuristic algorithms that can provide near-optimal solutions. Table I lists the main previous works that tackle this problem, categorized based on their objective, assumptions and solution techniques.

### A. Caching in non-congestible networks

Until some years ago, most of the research efforts in this area (e.g., [8], [9], [10], [11] to cite some of the most recognized) focused on scenarios where the bandwidth capacity of the network links always suffices to transport the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



TABLE I
RELATED WORKS ON JOINT CACHING AND ROUTING.

| Ref. | Objective | Topology | Congestible links | Technique | Solution |
|---|---|---|---|---|---|
| **Caching** | | | | | |
| [8] | Min delay | Arbitrary | ✗ | Relaxation & rounding | 10-approx. |
| [9] | Min delay | Hierarchical | ✗ | Swapping | 2-approx. |
| [10] | Min delay | 2-tier cell | ✗ | Submodularity | $\frac{e}{e-1}$-approx. |
| [11] | Min traffic cost | Arbitrary | ✗ | Submodularity | $\frac{e}{e-1}$-approx. |
| **Joint Caching and Routing** | | | | | |
| [16] | Min delay | 2-tier cell | Only one | Submodularity | $\frac{e}{e-1}$-approx. |
| [13] | Max cache hits | 2-tier cell | ✓ | Facility location | O(F)-approx. |
| [14] | Min schedule length | 2-tier cell | ✓ | Column generation | $(1+\epsilon)$-approx. |
| [15] | Max delay savings | 2-tier cell & devices | ✓ | Lagrangian relaxation | Heuristic |
| [12] | Max cache hits | Hierarchical IPTV | ✓ | Lagrangian relaxation | Heuristic |
| [24] | Min transmit power | 2-tier drone | ✓ | Learning & clustering | Heuristic |
| [25] | Min brown energy | Arbitrary | ✓ | Sequential fixing | Heuristic |
| [26] | Min traffic cost | 1-tier cell & devices | ✓ | Branch-and-bound | Exp.-time optimal |
| [27] | Min traffic energy | Arbitrary | ✓ | Branch-and-bound | Exp.-time optimal |
| [28] | Min delay | Arbitrary | ✓ | Conditional gradient | Heuristic |
| **Joint Caching, Routing and Computation** | | | | | |
| [29] | Max cache hits | 2-tier cell | ✓ | Submodularity | $\frac{1}{2}$-approx. |
| [30] | Max cache hits | 2-tier cell | ✓ | Randomized rounding | Bicriteria-approx. |
| [17] | Min cost & delay | Arbitrary | ✗ | BSUM | Heuristic |
| [18] | Min Energy & bandwidth | Arbitrary | ✓ | ADMM | Heuristic |
| **This work** | Max cache hits | Arbitrary | ✓ | Randomized rounding | $\frac{e}{e-1}\left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}$-approx. |

contents from the caches to the end-users (non-congestible links). Under this assumption, the question of routing the content requests to caches becomes trivial; simply routes each request to the lowest-delay or lowest-cost cache having stored the requested content, depending on the objective of interest. Here, the delay or cost were modeled as linear functions of the number of hops between the caches and the end-users. This simplification allowed the derivation of tight approximations for this problem; a 10-approximation using linear-relaxation and rounding techniques in [8], a 2-approximation combinatorial algorithm that iteratively swaps files in and out of the caches in [9], as well as other improved $e/(e-1)$-approximations based on the submodularity property of this problem [10], [11].

### *B. Joint caching and routing in congestible networks*

Interestingly, the caching problem becomes more challenging if we take into account the bandwidth capacities of the (congestible) links in the network (e.g., for small network operators that cannot provision enough capacity or for the scenarios of massive demand). In this case, the content caching and request routing decisions affect each other and therefore they need to be optimized jointly. Due to the high complexity of this joint problem, recent efforts have focused on special network topologies that facilitate its solution. Dehghan et al. [16] formulated this problem for a two-tier network consisting of a remote server and many local caches in proximity to end-users. For the case that only the link to the remote server is congestible, this problem was formulated as a submodular problem and an $e/(e-1)$-approximation was derived. However, as we showed in Section III-B, the submodularity property does not hold for the general case of arbitrary network topologies and multiple congestible links. For a similar 2-tier network setup (where caches are installed at small-cell base stations to offload a macro-cell base station), Poularakis et al. [13] proposed facility-location inspired algorithms with approximation ratios that in the worst case increase proportionally with the number of files. However, the reduction to the facility location problem cannot be extended for networks of arbitrary topologies. This problem was extended to account for the interference caused by the base station data transmissions in [14]. An approximation algorithm was developed using the column generation method that approaches the optimal solution in exponential time. Another extension was provided in [15] by allowing the caching of contents at the user devices. A heuristic algorithm was proposed based on the Lagrangian relaxation method. The same method was used in [12] for the video caching in a hierarchical IPTV network scenario. Additional heuristic algorithms have been proposed for different 2-tier network scenarios. For a 2-tier network formed by drones and infrastructure cache-nodes, learning and clustering techniques were applied in [24]. For cache-nodes that operate using renewable energy a sequential-fixing algorithm was proposed in [25]. Another integer programming formulation was proposed in [26] for the scenario of caching contents at a cellular base station and the mobile devices. Problems of this type can be solved using branch and bound integer solvers, but these solvers require exponential time and thus do not scale for large problem instances. A similar formulation was proposed in [27] to minimize the energy consumption caused by the traffic transmission in networks of arbitrary topology. For arbitrary topologies, a conditional gradient-based heuristic method

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



was also proposed in [28]. However, all the aforementioned works either assume special topologies (2-tier, hierarchical) or propose exponential-time or heuristic methods without any approximation guarantees.

### C. Joint caching, routing and computation

Additional joint caching and routing schemes have been proposed in [29], [30], [17], [18]. These works consider the placement or caching of *services* that require not only storage and bandwidth resources but also *computation* resources for some task execution associated with the service in mobile-edge computing for IoT. We note that these joint caching, routing and computation algorithms can be used to solve the joint caching and routing problem as well since they solve a more general problem. However, the existing solution approaches in [29], [30], [17], [18] have the same limitations with all the approaches mentioned above (i.e., either solve the problem for a special topology or find heuristic solutions). Our work in this paper fills this gap in the literature by designing approximation algorithms for congestible networks of arbitrary topology.

## III. System Model and Problem Formulation

### A. System model

We consider a heterogeneous network consisting of a large number of 5G and IoT interconnected nodes such as BSs, femto-cells, edge cloud servers, fog nodes, gateways, etc. We represent the network by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ and $\mathcal{L}$ indicate the sets of nodes and links, respectively. A subset of the nodes equipped with caches is denoted by $\mathcal{N}_c \subseteq \mathcal{N}$. A set of users $\mathcal{K}$ arbitrarily distributed over the network generate requests for content files (e.g., video clips, sensing data, etc.) as illustrated in Figure 1.

We denote by $\mathcal{F}$ the set of possible content files, also referred by the library. All files in the library have the same size normalized to one. This is a mild assumption that can be easily removed as, in real systems, files can be divided into chunks or blocks of the same size (e.g., see the discussion in [10], [11], [12], [15]). Each user $k \in \mathcal{K}$ requests one of these files denoted by $f(k) \in \mathcal{F}$[1]. The requests can be predicted by the network operator by using historical traffic data and applying machine learning techniques [21], [31].

For each file request of a user, the network operator needs to find an in-network cache that stores and delivers the file through the network along some routing paths, possibly of multiple hops. Each file request successfully served will consume the bandwidth of every link along the routing path. These routing paths should be carefully chosen so that network congestion can rarely happen. Additionally, since the data size of a request is usually much smaller than that of a file, we ignore the respective bandwidth consumption to deliver the request and merely count the traffic volume for content data.

[1]We note that a real-life user requesting more than one files can be simply represented by multiple users in the set $\mathcal{K}$.

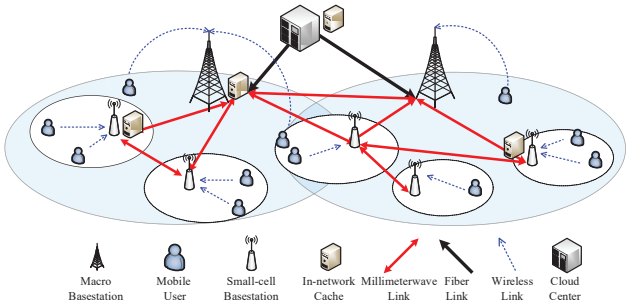


Fig. 1. An example network. Four out of eight nodes are equipped with caches that can be used to store and deliver content to users upon requests. Link capacities limit the volume of requests routed from caches to users.

Both of the links and the nodes in the network are capacitated, i.e., the bandwidth capacity of a link $l \in \mathcal{L}$ and the storage capacity of a node $n \in \mathcal{N}$ are limited. We denote the bandwidth capacity by $B_l$ as the maximum number of files a link $l$ can deliver during a given time period of interest (e.g., a few hours). Also, the maximum number of files that a node $n \in \mathcal{N}_c$ can store is denoted by $C_n$.

For each pair of a user $k \in \mathcal{K}$ and a cache-node $n \in \mathcal{N}_c$, there may be several possible paths that can be used for file delivery. We denote this set of possible paths by $\mathcal{P}_{kn}$. For example, $\mathcal{P}_{kn}$ may include all the paths connecting $k$ and $n$ that consist of at most a given number of hops or within a maximum end-to-end delay budget. This way, we prevent users from experiencing prohibitively large content delivery delays.

### B. Problem formulation

We introduce the optimization variable $x_n^f \in \{0, 1\}$ that indicates whether the cache of node $n$ has stored file $f$ ($x_n^f = 1$) or not ($x_n^f = 0$). The *caching policy* is then described by the matrix:

$$\mathbf{x} = (x_n^f : \ n \in \mathcal{N}_c, f \in \mathcal{F}). \tag{1}$$

We also introduce the optimization variable $y_{kn}^p \in \{0, 1\}$ that indicates whether the file request of user $k$ is routed to the cache-node $n$ along the path $p$ ($y_{kn}^p = 1$) or not ($y_{kn}^p = 0$). The respective *routing policy* matrix is denoted by the matrix:

$$\mathbf{y} = (y_{kn}^p : \ k \in \mathcal{K}, n \in \mathcal{N}_c, p \in \mathcal{P}_{kn}). \tag{2}$$

Due to the limited cache sizes, it may not be possible to store all the files in the caches. Even if this is possible, the bandwidth capacity of the links may not suffice to serve all the users by the caches. The goal of the network operator is to find the caching and routing policies $\mathbf{x}$ and $\mathbf{y}$ that maximize the volume of file requests that can be served by the caches. Formally, the *Joint Caching and Routing in congestible networks of Arbitrary Topology (JoCRAT)*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



*problem* can be expressed as follows:

$$\max_{\mathbf{x},\mathbf{y}} \sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn}} y_{kn}^p \tag{3}$$

$$\text{s.t.:} \sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn},p\ni l} y_{kn}^p \leq B_l,\ \forall l\in\mathcal{L}, \tag{4}$$

$$\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn}} y_{kn}^p \leq 1,\ \forall k\in\mathcal{K}, \tag{5}$$

$$y_{kn}^p \leq x_n^{f(k)},\ \forall k\in\mathcal{K}, n\in\mathcal{N}_c, p\in\mathcal{P}_{kn}, \tag{6}$$

$$\sum_{f\in\mathcal{F}} x_n^f \leq C_n,\ \forall n\in\mathcal{N}_c, \tag{7}$$

$$x_n^f\in\{0,1\}, y_{kn}^p\in\{0,1\}, \tag{8}$$
$$\forall n\in\mathcal{N}_c, f\in\mathcal{F}, k\in\mathcal{K}, p\in\mathcal{P}_{kn},$$

where the objective (3) stands for the total volume of requests served by the caches (cache hits). Inequalities in (4) represent the bandwidth constraints on the links. Inequalities in (5) ensure that each user request will be served at most once. Inequalities in (6) indicate that node $n$ can serve the file request of user $k$ if $f(k)$ is stored in the cache of node $n$. The cache size constraints are represented by (7). Finally, constraints in (8) indicate the integer nature of the optimization variables.

JoCRAT is an integer optimization problem, and this type of problems are typically hard to solve. In fact, JoCRAT generalizes several NP-Hard caching problems in literature. For example, JoCRAT degrades into the caching problem described in [13] when the graph $\mathcal{G}$ is bipartite (2-tier) and the independent nodes are the $\mathcal{N}$ and $\mathcal{K}$ sets. JoCRAT also degrades into the caching problem described in [12] when $\mathcal{G}$ is hierarchical. Therefore, the following theorem can be shown.

*Theorem 1:* The JoCRAT problem is NP-Hard.

In the next two sections, we study the complexity of the JoCRAT problem further and present efficient solutions.

## IV. Complexity Analysis

In this section, we show that for the special case of one cache the JoCRAT problem can be approximatively solved using existing algorithms in the literature. This result is based on the submodularity property of the objective function, a property that has been used by several works in the past to tackle various caching problems. This attractive property, however, does not hold for the general case of our problem.

### A. Special case of one cache

We consider the special case where there is only one cache-node in the network, i.e., $|\mathcal{N}_c|$=1. Yet, we make no restrictions on the network paths that can be used for the delivery of content, i.e., all the possible network paths are included in the $\mathcal{P}_{kn}$ set. We note that the JocRAT problem remains challenging even in this special case. Consider for example the network in Figure 1 when there is only one cache deployed at the leftmost BS. If the bandwidth capacities were abundant (very large values $B_l, \forall l\in\mathcal{L}$), then finding the optimal caching policy would be trivial; simply cache the most popular files with respect to the demand of *all* the end-users. However, if the links connecting the two BSs have arbitrarily low bandwidth capacities, it would be wasteful to cache any of the files requested by the end-users in the rightmost macrocell (since these end-users cannot be served by the cache in any case). Instead, the most popular files with respect to the demand of the end-users in the leftmost BS only should be cached. Therefore, the bandwidth capacities make unclear what is the optimal caching policy and the problem is far from trivial.

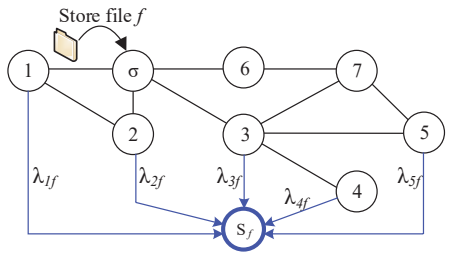


Fig. 2. Reduction from JoCRAT to SLP problem. The instance of the SLP problem is depicted that is equivalent to the JoCRAT instance in Figure 1 for the special case of only one cache deployed at the leftmost BS.

We will show that in this special case the JoCRAT problem falls into the class of submodular maximization problems. Formally, we introduce the definition of submodular functions.

*Definition 1:* Given a finite set of elements $\mathcal{G}$ (referred to as ground set) a set function $\phi: 2^{\mathcal{G}}\rightarrow\mathbb{R}$ is submodular if for any sets $\mathcal{X}_1\subseteq\mathcal{X}_2\subseteq\mathcal{G}$ and every element $g\in\mathcal{G}\setminus\mathcal{X}_2$, it holds:

$$\phi(\mathcal{X}_1\cup\{g\})-\phi(\mathcal{X}_1)\geq\phi(\mathcal{X}_2\cup\{g\})-\phi(\mathcal{X}_2), \tag{9}$$

i.e., the marginal value of the function when adding a new element in a set decreases as this set expands.

We will prove the submodularity property for the objective function of our JoCRAT problem, by showing that our problem is equivalent to the *Sink Location Problem (SLP)* [32].

*SLP:* We are given a capacitated network with a single source $\sigma$, a set of possible sinks $\mathcal{S}$ and a number $Q>0$. The question is to locate a subset $\mathcal{X}\subseteq\mathcal{S}$ of $Q$ sinks so as to maximize the flow that can be sent from the source to the located sinks.

*Lemma 1:* For the special case of one cache, JoCRAT is equivalent to the SLP problem.

*Proof:* To show the equivalence of the two problems, consider the example in Figure 2. Given the instance of the JoCRAT problem, we can construct the equivalent instance of the SLP problem as follows. First, we create a capacitated network with the same topology as in the JoCRAT instance. Second, we install a source $\sigma$ at the node corresponding to the cache-node in the JoCRAT instance. Third, we create a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



sink $s_f$ for each file $f \in \mathcal{F}$ and define the respective set of sinks $\mathcal{S}$ accordingly. We connect each sink $s_f$ to every node $n$ in the network with a link of capacity equal to the total user demand at node $n$ for file $f$, denoted by $\lambda_{nf}$. It is not difficult to show that storing a file $f$ in the cache in the JoCRAT instance is equivalent to locating a sink $s_f$ in the SLP instance. Similarly, storing a subset of files in the cache is equivalent to locating the respective subset of sinks. The maximum flow in the SLP instance will be equal to the maximum number of cache hits in the JoCRAT instance. ■

Given a subset of sinks $\mathcal{X} \subseteq \mathcal{S}$, we define by $g(\mathcal{X})$ the maximum flow in the SLP instance. Previous works have shown that the function $g(\mathcal{X})$ is submodular (see Proposition 3.3 in [32]). Due to the equivalence of the two problems, the submodularity property holds for the objective of cache hits as well. This is an important result since there exist several approximation algorithms in the literature for maximizing a monotone submodular function. For example, a simple greedy algorithm that iteratively stores the file in the cache that improves the objective function the most, until the cache becomes full, returns a solution with value at most $e/(e-1) \simeq 1.58$ times worse than the optimal [20]. Formally, the following theorem holds.

*Theorem 2:* For the special case of one cache, there exists an $e/(e-1)$-approximation algorithm for the JoCRAT problem.

### B. General case

It would be tempting to conjecture that the property of submodularity holds for the general case of the JoCRAT problem. However, as we show below, this attractive property does not hold when more than one caches are available. Specifically, consider the counter-example of a wireless caching network in Figure 3. There are two BSs each one equipped with a cache, and four end-users inside the intersection of the coverage regions of the BSs. Hence, there exists a single one-hop routing path from each user to each cache. The first two end-users request file $f_1$, while the rest two end-users request file $f_2$. For the first BS, the cache size is $C_1 = 1$ and the bandwidth capacity of its downlink is $B_1 = 2$. For the second BS, we set $C_2 = 2$ and $B_2 = 2$.

To show that the submodularity property does not hold for this example, we consider two caching policies represented by the sets $\mathcal{X}_1 = \{x_2^1\}$ where BS 2 caches file 1 and $\mathcal{X}_2 = \{x_2^1, x_2^2\}$ where BS 2 caches both files 1 and 2. Clearly, $\mathcal{X}_1 \subseteq \mathcal{X}_2$. The set function $g(\mathcal{X})$ is used to represent the cache hits for a given caching policy $\mathcal{X}$. The number of cache hits for the two caching policies will be: $g(\mathcal{X}_1) = 2$ and $g(\mathcal{X}_2) = 2$ since in both cases the bandwidth capacity of BS 2 limits the number of served end-users to two. Next we consider the number of cache hits when we add the element $\{x_1^1\}$ to the two sets. It becomes $g(\mathcal{X}_1 \cup \{x_1^1\}) = 2$ and $g(\mathcal{X}_2 \cup \{x_1^1\}) = 4$ since in the $\mathcal{X}_1 \cup \{x_1^1\}$ policy only the two end-users requesting file $f_1$ can be served. Therefore, it holds that:

$$g(\mathcal{X}_2 \cup \{x_1^1\}) - g(\mathcal{X}_2) > g(\mathcal{X}_1 \cup \{x_1^1\}) - g(\mathcal{X}_1) \quad (10)$$

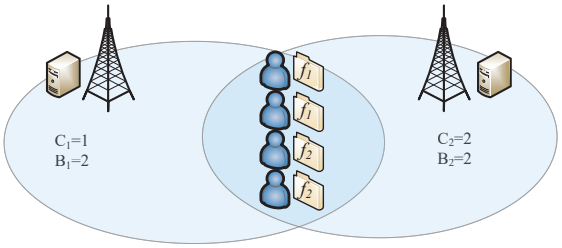


Fig. 3. A counterexample showing that the objective function of JoCRAT is not submodular in the general case.

In other words, the marginal gain of adding element $x_1^1$ is higher for set $\mathcal{X}_2$ than $\mathcal{X}_1$, and $\mathcal{X}_2 \supset \mathcal{X}_1$. This contradicts the definition of submodular functions.

The cases described above help us to obtain a better understanding of the complexity of JoCRAT problem and which of the previous results in the literature can be exploited to tackle it. However, the problem in its general form remains open. In the next section, we will show how to address it.

## V. Approximation to JoCRAT problem

In this section, we present one of the main contributions of this work, a novel joint caching and routing algorithm that achieves an approximate solution to the JoCRAT problem. The approximation ratio of this algorithm is *sublinear* on the number of nodes and links in the network. To the best of our knowledge, this is the first non-trivial approximation for this important problem. We summarize this result in the following theorem.

*Theorem 3:* Define $M = \min_{n \in \mathcal{N}_c, l \in \mathcal{L}} \{C_n, B_l\} > 2$ and $R = |\mathcal{N}_c| + |\mathcal{L}|$. Then, there exists a polynomial-time algorithm that finds a feasible solution to the JoCRAT problem with value at most $(\frac{e}{e-1}) \times \left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}$ times lower than the optimal.

We defer the proof of the above theorem to the Appendix B. The proposed algorithm builds upon *randomized rounding*, a popular technique for deriving approximate solutions to various NP-Hard problems in literature, e.g., the unsplittable flow routing [33] and general assignment [34] problems, to name two. The algorithms of this type typically start by solving a relaxation of the problem and then round the variables from fractional to integer values. The rounding may be performed in iterations over which the value of the objective (e.g., in-network flow volume) increases while at the same time the risk of violating constraints (e.g., link bandwidth capacities) is kept low. As we explain in the sequel, however, our problem is more complicated than the aforementioned problems as it involves both caching and routing decisions, which are interrelated (cf. constraint (6)), and thus a new variant of the randomized rounding algorithm is required.

In the rest of this section, we describe in detail the proposed algorithm, referred to as JoCRAT algorithm. We begin

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



by describing the relaxation of the problem (Subsection V-A) and the risks of violating the constraints (Subsection V-B) that will be later used for the algorithm description (Subsection V-C).

### A. Linear Relaxation of JoCRAT problem

We allow the optimization variables in $\mathbf{x}$ and $\mathbf{y}$ to take any real value from 0 to 1, as opposed to integer values only:

$$x_n^f \in [0,1], y_{kn}^p \in [0,1],\ \forall n \in \mathcal{N}_c, f \in \mathcal{F}, k \in \mathcal{K}, p \in \mathcal{P}_{kn}. \tag{11}$$

Formally, the *Linear Relaxation of JoCRAT problem (LR-JoCRAT)* is expressed as follows:

$$\max_{\mathbf{x},\mathbf{y}} \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} y_{kn}^p \tag{12}$$
$$\text{s.t.: constraints: } (4),(5),(6),(7),(11).$$

LR-JoCRAT problem can be optimally solved using standard linear optimization techniques. We denote by $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ the optimal (fractional) solution values to this problem.

### B. Risks of Rounding

The randomized rounding algorithm we propose rounds the fractional values in $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ into binary values $\mathbf{x}^*$ and $\mathbf{y}^*$. If not appropriately designed, however, this rounding process risks to violate the constraints of the problem. In addition, the obtained objective function value (cache hits) risks to be out of the approximation region specified in Theorem 3. We dedicate this subsection to formally define these risks which will be later used in the description of the algorithm and the proof of Theorem 3.

We define the events $\mathbf{E}_l$ and $\mathbf{E}_n$ that the bandwidth capacity of link $l \in \mathcal{L}$ and the cache capacity of node $n \in \mathcal{N}$ are violated after rounding, respectively. Similarly, we define by $\mathbf{E}_{ch}$ the event that the number of cache hits after rounding is more than $(\frac{e}{e-1}) \times \left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}$ times worse than the optimal. This essentially means that the approximation ratio of Theorem 3 is not reached.

The proposed algorithm rounds the variables in $\bar{\mathbf{y}}$ *sequentially* over the end-users in the set $\mathcal{K}$. Therefore, we define the above events conditioned to the values rounded so far. Specifically, we define by $\mathcal{U} \subseteq \mathcal{K}$ the subset of end-users for which the rounding decisions have been made so far and by $\mathbf{y}^*(\mathcal{U})$ the respective rounded variable vector. We next define the *conditional probabilities* $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\}$, $\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}$ and $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$ that the respective events occur given the rounded variable vector $\mathbf{y}^*(\mathcal{U})$. To compute these conditional probabilities (or, *risks*), we will use the results in the following lemma.

*Lemma 2 (Poisson Binomial Distribution [35]):* Let $y_t$, $t = 1,\cdots,T$ be a series of $T$ independent and non-identical distributed random indicators. In particular, $y_t \sim$ Bernoulli$(\bar{y}_t), t = 1,\cdots,T$, where $\bar{y}_t$ is the success probability of indicator $y_t$. Then, the summation $\sum_{t=1}^{t=T} y_t$ is defined by a Poisson Binomial random variable, and its closed-form CDF is given by the following Discrete Fourier Transformation (DFT):

$$\mathbb{P}\{\sum_{t=1}^{t=T} y_t \le B\} = \sum_{k=0}^{k=B} \left\{ \frac{1}{T+1} \sum_{l=0}^{l=T} W^{-lk} \prod_{t=1}^{t=T} \left[1 - \bar{y}_t + \bar{y}_t W^l\right] \right\} \tag{13}$$

where $B \le T$ is a positive constant, $W = \exp(\frac{2\pi\mathbf{i}}{T+1})$ is the DFT core, and $\mathbf{i}$ stands for the imaginary unit in complex numbers.

We note that if $B > T$ in the above expression, the respective probability is trivially equal to one since even if all the random variables take the value one they cannot exceed the threshold value $B$. By using the above lemma, we will derive the expressions for the aforementioned risks as follows.

**1) $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\}$ risk expression**. For a given $\mathbf{y}^*(\mathcal{U})$, the end-users in $\mathcal{U}$ have already consumed $\sum_{k\in\mathcal{U}} \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}:l\in p} y^{*p}_{kn}$ bandwidth capacity of link $l$. Thus, the bandwidth that remains for the end-users not in $\mathcal{U}$ is equal to $B_l - \sum_{k\in\mathcal{U}} \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}:l\in p} y^{*p}_{kn}$. Therefore, we obtain that:

$$\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\} = 1 - \mathbb{P}\left\{ \sum_{k\notin\mathcal{U}} \Big[ \sum_{n\in\mathcal{N}_c} \sum_{\substack{p\in\mathcal{P}_{kn} \\ l\in p}} y_{kn}^p \Big] \le B_l - \sum_{k\in\mathcal{U}} \Big[ \sum_{n\in\mathcal{N}_c} \sum_{\substack{p\in\mathcal{P}_{kn} \\ l\in p}} y^{*p}_{kn} \Big] \right\} \tag{14}$$

Next, we define the number of end-users that have already routed their requests through link $l$:

$$b_l(\mathcal{U}) \triangleq \sum_{k\in\mathcal{U}} \Big[ \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}:l\in p} y^{*p}_{kn} \Big]. \tag{15}$$

and the number of end-users that may route their requests through link $l$ in next iterations of the algorithm as:

$$q_l(\mathcal{U}) \triangleq \sum_{k\notin\mathcal{U}} [1 - \prod_{n\in\mathcal{N}_c} \prod_{p\in\mathcal{P}_{kn}} (\mathbf{1}_{l\notin p})]. \tag{16}$$

The above expression counts the end-users that have not been examined yet by the algorithm and for which there is at least one path available for them that contains link $l$.

We note that for each user $k \notin \mathcal{U}$ the summation $\sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}:l\in p} y_{kn}^p$ represents a random variable. This random variable is binary because of the constraint (5). Besides, these variables are independent across end-users because of the way the algorithm assigns values to them (as we will explain in the next subsection). The respective success probability is equal to $\sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}:l\in p} \bar{y}_{kn}^p$. Therefore, we can apply Lemma 2 for $B = B_l - b_l(\mathcal{U})$ and $T = q_l(\mathcal{U})$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



and, assuming that $B_l - b_l(\mathcal{U}) \leq q_l(\mathcal{U})$, to obtain that[2]:

$$\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\} = 1 - \sum_{s=1}^{B_l-b_l(\mathcal{U})} \Big\{ \frac{1}{1+q_l(\mathcal{U})} \sum_{m=0}^{m=q_l(\mathcal{U})} W_l^{-sm} \Big[ \prod_{k\notin\mathcal{U}} (1 - \big[ \sum_{n\in\mathcal{N}_c} \sum_{\substack{p\in\mathcal{P}_{kn} \\ l\in p}} \bar{y}_{kn}^p \big] + \big[ \sum_{n\in\mathcal{N}_c} \sum_{\substack{p\in\mathcal{P}_{kn} \\ l\in p}} \bar{y}_{kn}^p \big] W_l^m) \Big] \Big\}, \tag{17}$$

where $W_l \triangleq \exp(\frac{2\pi\mathbf{i}}{q_l(\mathcal{U})+1})$ stands for the corresponding DFT core. We note that when $B_l - b_l(\mathcal{U}) > q_l(\mathcal{U})$, the end-users that have not been examined yet by the algorithm are not many enough to violate the capacity of link $l$ and, therefore, $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\}$ is trivially equal to zero.

**2)** $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$ **risk expression**. The analysis of $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$ is similar to the $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\}$ expression. For a given $\mathbf{y}^*(\mathcal{U})$, the service of the end-users in the set $\mathcal{U}$ has already ensured $\sum_{k\in\mathcal{U}} \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} y^{*p}_{kn}$ cache hits. The approximation ratio of Theorem 3 will not be reached, however, if the additional cache hits associated to the end-users not in $\mathcal{U}$ are not enough relatively to the optimal solution value (OPT). Since, we do not know the value of OPT, we will use in our analysis the value of the linear relaxed problem (LOPT), which upper bounds OPT. Specifically, the number of additional cache hits that the algorithm should achieve to ensure that the approximation ratio holds is equal to:

$$\text{APX} = \frac{\text{LOPT}}{(\frac{e}{e-1}) \times \left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}} - \sum_{k\in\mathcal{U}} \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} y^{*p}_{kn}. \tag{18}$$

Therefore, we obtain that:

$$\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\} = \mathbb{P}\Big\{ \sum_{k\notin\mathcal{U}} \big[ \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} y^p_{kn} \big] \leq \lfloor \text{APX} \rfloor \Big\} \tag{19}$$

By applying Lemma 2, we obtain that:

$$\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\} = \sum_{s=1}^{\lfloor \text{APX} \rfloor} \Big\{ \frac{1}{1+q_{ch}(\mathcal{U})} \sum_{m=0}^{m=q_{ch}(\mathcal{U})} W_{ch}^{-sm} \Big[ \prod_{k\notin\mathcal{U}} (1 - \big[ \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} \bar{y}_{kn}^p \big] + \big[ \sum_{n\in\mathcal{N}_c} \sum_{p\in\mathcal{P}_{kn}} \bar{y}_{kn}^p \big] W_{ch}^m) \Big] \Big\}, \tag{20}$$

where $q_{ch}(\mathcal{U}) \triangleq |\mathcal{K}\backslash\mathcal{U}|$, and $W_{ch} \triangleq (\exp \frac{2\pi\mathbf{i}}{q_{ch}(\mathcal{U})+1})$ stands for the corresponding DFT core. We note that when $\lfloor \text{APX} \rfloor > q_{ch}(\mathcal{U})$, the end-users that have not been examined yet are not many enough to reach the approximation ratio and, therefore, $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$ is trivially equal to one.

**3)** $\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}$ **risk expression**. Compared to $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\}$ and $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$, analyzing $\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}$ requires some more effort. Specifically, in order for constraint (6) to always hold, the algorithm will round the caching variables using the following rule:

$$x^{*f}_n = 1 - \prod_{k:f(k)=f} \big[1 - \sum_{p\in\mathcal{P}_{kn}} y^{*p}_{kn}\big], \forall n \in \mathcal{N}_c, \forall f \in \mathcal{F}. \tag{21}$$

The above means that $x^{*f}_n$ will be 1 if at least one user requesting file $f$ is served by cache $n$. We note that $\big[1 - \sum_{p\in\mathcal{P}_{kn}} y^p_{kn}\big]$ is a binary random variable since user $k$ can only fetch a file from a single cache-node and through a single path. Besides, these binary random variables are independent for different end-users because of the way the proposed algorithm assigns values to them. Therefore, $x^f_n$ terms are also independent binary random variables, the success probabilities of which are given by:

$$\bar{x}^f_n \triangleq 1 - \prod_{k\notin\mathcal{U}:f(k)=f} \big[1 - \sum_{p\in\mathcal{P}_{kn}} \bar{y}^p_{kn}\big], \forall n \in \mathcal{N}, f \in \mathcal{F}. \tag{22}$$

By applying Lemma 2, we obtain that:

$$\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\} = 1 - \sum_{s=1}^{C_n-b_n(\mathcal{U})} \Big\{ \frac{1}{1+q_n(\mathcal{U})} \times \sum_{m=0}^{m=q_n(\mathcal{U})} W_n^{-sm} \Big[ \prod_{k\notin\mathcal{U}:f(k)=f} (1 - \bar{x}^f_n + \bar{x}^f_n W_n^m) \Big] \Big\}, \tag{23}$$

where

$$b_n(\mathcal{U}) \triangleq \sum_{f\in\mathcal{F}} \big[1 - \prod_{k\in\mathcal{U}:f(k)=f} \big(1 - \sum_{p\in\mathcal{P}_{kn}} y^{*p}_{k,n}\big)\big] \tag{24}$$

represents the number of files that have been stored in the cache of node $n$ so far, and

$$q_n(\mathcal{U}) \triangleq \sum_{k\notin\mathcal{U}} \mathbf{1}_{\{\mathcal{P}_{kn}\neq\emptyset\}}. \tag{25}$$

represents the number of files that may be stored at cache $n$ in next iterations of the algorithm. Here, $\mathbf{1}_{\{.\}}$ is the indicator function. $W_n \triangleq \exp(\frac{2\pi\mathbf{i}}{q_n(\mathcal{U})+1})$ denotes the corresponding DFT core. Similar to the previous expressions, we note that when $C_n - b_n(\mathcal{U}) > q_n(\mathcal{U})$, the end-users that have not been examined yet are not many enough to violate the capacity of cache $n$ and, therefore, $\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}$ is trivially equal to zero.

**4) Aggregate risk expression**. The last step is to evaluate the aggregate risk that *any* of the events $\mathbf{E}_l, \forall l \in \mathcal{L}$, $\mathbf{E}_n, \forall n \in \mathcal{N}_c$ and $\mathbf{E}_{ch}$ will happen. Since this is extremely challenging to compute [36], we find instead an upper bound on the aggregate risk, described in the following lemma (proved in the Appendix A).

*Lemma 3 (Potential Energy (PE)):* Given an instance of the JoCRAT problem, we define the PE function for a given $\mathbf{y}^*(\mathcal{U})$ as follows:

$$h(\mathbf{y}^*(\mathcal{U})) = 1 + \mathbb{P}\big\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\big\} - \prod_{l\in\mathcal{L}} \big(1 - \mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\big) \times \prod_{n\in\mathcal{N}_c} \big(1 - \mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\big) \tag{26}$$

Then, PE upper bounds the aggregate risk of violating any of the bandwidth and cache capacity constraints

[2]Here, we use the convention that both the results of $\sum_{s\in\emptyset}$ and $\sum_{s=1}^{s=a-b}$ where $a < b$ are zeros, and that the result of $\Pi_{k\in\emptyset}$ is one.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



or the number of cache hits being less than LOPT $\times$ $(1-\frac{1}{e}) \times \left(\frac{e-1}{2Re^{M+1}}\right)^{\frac{1}{M-2}}$.

### C. The proposed JoCRAT algorithm

Having defined the risk and PE expressions, we can now formally present the JoCRAT algorithm. The pseudocode is given in Algorithm 1. It consists of three phases, as described in detail below.

**A. Initialization phase**. JoCRAT algorithm first solves the linear relaxed problem to find the initial fractional solutions $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ (line 2). It also initializes $\mathcal{U}$ to $\emptyset$ and $\mathbf{y}^*(\emptyset)$ to the all-zero vector (line 3). Then, it scales down the initial values of $\bar{\mathbf{y}}$ by a factor $\gamma$. The $\gamma$ value is picked such that the initial value of the PE function $h(\mathbf{y}^*(\emptyset))$ is close to but *strictly* less that one. Satisfying this condition at the initialization phase is critical for ensuring that undesirable events will not occur in the end of the algorithm, as described in the proof of Theorem 3. In order to find such a $\gamma$ value the algorithm performs a bisection search (lines 4-6). This search starts with $\gamma = e$ and will increase it as much as needed for the PE function to become strictly less than one. We note that as the $\gamma$ value changes during the search, the $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}$ also change. Specifically, $\bar{\mathbf{y}}$ is scaled down by its initial value by the current factor $\gamma$, while $\bar{\mathbf{x}}$ is updated according to the rule in equation (22). This update will affect the risk values since they depend on $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ (cf. equations (17), (20), (23)), which in turn will affect the PE function value (cf. equation (26)).

**B. Sequential rounding phase**. In this phase, the algorithm iteratively examines all the end-users one-by-one to decide how to compute the rounded routing variables $y^{*p}_{kn}$ $\forall k, n, p$ (lines 8-17). At a typical stage of the algorithm, we are examining some user $k$, where $k \notin \mathcal{U}$. The end-users in $\{1, 2, \ldots, k-1\} \triangleq \mathcal{U}$ have been already examined and the respective routing variables have been already rounded. For the currently examined user $k$, the algorithm checks if there is a cache $n$ and a path $p \in \mathcal{P}_{kn}$ to route the request of user $k$, or equivalently round $y^{*p}_{kn}$ to one, such that the PE function value will remain less than one (line 9). If yes, we round $y^{*p}_{kn}(\mathcal{U} \cup \{k\}) = 1$ and $y^{*p'}_{kn'}(\mathcal{U} \cup \{k\}) = 0$ $\forall n' \neq n$ and $p' \neq p$ (lines 10-11). If not, we round $y^{*p'}_{kn'}(\mathcal{U} \cup \{k\}) = 0$ $\forall n' \in \mathcal{N}_c$ and $p' \in \mathcal{P}_{kn}$ (the request of user $k$ is not served) (line 13). The rest values of the vector $y^*(\mathcal{U} \cup \{k\})$ are the same with the vector $y^*(\mathcal{U})$ (line 15). In the end of the loop, i.e., when $\mathcal{U} = \mathcal{K}$, we will have computed the final rounded values of the routing variables $\mathbf{y}^* = \mathbf{y}^*(\mathcal{K})$ (line 18).

**C. Final rounding phase**. The last step is to round the caching variables $\mathbf{x}^*$ based on the values in $\mathbf{y}^*$. This will be done by applying the rule in equation (21) (lines 20-22).

Finally, we analyze the computational complexity of JoCRAT algorithm. From equation (17), (20), (23), and (26), the complexity to evaluate the PE function is $\mathcal{O}\big((|\mathcal{L}| + |\mathcal{N}_c|) \times |\mathcal{N}_c| \times |\mathcal{K}| \times \log(|\mathcal{N}_c| \times |\mathcal{K}|)\big)$. Since the proposed algorithm calculates the PE function at most $\mathcal{O}(|\mathcal{N}_c| \times |\mathcal{K}|)$ times ($|\mathcal{N}_c|$ times for each user), the overall complexity is $\mathcal{O}\big((|\mathcal{L}| + |\mathcal{N}_c|) \times |\mathcal{N}_c|^2 \times |\mathcal{K}|^2 \times \log(|\mathcal{N}_c| \times |\mathcal{K}|)\big)$. This complexity is acceptable in practice given that the problem will be solved *offline* by the network operator, i.e., in the beginning of each day using predictions of the user demand. In the next section, we perform evaluations that show the running time in practical scenarios.

**Algorithm 1** JoCRAT

1: **A. Initialization phase:**
2: Solve the LR-JoCRAT problem to find $\bar{\mathbf{x}} = \{\bar{x}^1_1, \cdots, \bar{x}^f_n, \cdots\})$ and $\bar{\mathbf{y}} = \{\bar{y}^1_{11}, \cdots, \bar{y}^p_{kn}, \cdots\}$.
3: Init $\mathcal{U} = \emptyset$ and $\mathbf{y}^*(\emptyset) = \mathbf{0}$.
4: **repeat**
5: Bisection Search: $\gamma \in \left(e, \left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}\right]$
6: **until** $1 - \epsilon \leq h(\mathbf{y}^*(\emptyset)) < 1$
7: **B. Sequential rounding phase:**
8: **for** $k \in \mathcal{K}$ **do**
9: **if** $\exists\ n \in \mathcal{N}_c$ and $p \in \mathcal{P}_{kn}$: $h(\mathbf{y}^*(\mathcal{U} \cup \{k\})) < 1$ **then**
10: Round $y^{*p}_{kn}(\mathcal{U} \cup \{k\}) = 1$
11: Round $y^{*p'}_{kn'}(\mathcal{U} \cup \{k\}) = 0\ \forall n' \neq n$ and $p' \neq p$
12: **else**
13: Round $y^{*p'}_{kn'}(\mathcal{U} \cup \{k\}) = 0\ \forall n', p'$
14: **end if**
15: Set $y^{*p'}_{k'n'}(\mathcal{U} \cup \{k\}) = y^{*p'}_{k'n'}(\mathcal{U})\ \forall k' \neq k, n', p'$
16: $\mathcal{U} = \mathcal{U} \cup \{k\}$
17: **end for**
18: $\mathbf{y}^* = \mathbf{y}^*(\mathcal{U})$
19: **C. Final rounding phase:**
20: **for** $n \in \mathcal{N}$,$f \in \mathcal{F}$ **do**:
21: $x^{*f}_n = 1 - \prod_{k:f(k)=f} \left[1 - \sum_{p \in \mathcal{P}_{kn}} y^{*p}_{k,n}\right]$
22: **end for**
23: **return** $\mathbf{x}^*$ and $\mathbf{y}^*$

## VI. Evaluation Results

In this section, we conduct extensive evaluations to demonstrate the advantages of the proposed JoCRAT algorithm over existing JCR algorithms. Various IoT scenarios differing in the network topologies, available bandwidth and storage resources, and file requests patterns are examined. Overall, we find that JoCRAT can increase the cache hits especially in the low and moderate bandwidth capacity regimes. In the rest of this section, we discuss these results in detail.

### A. Compared JCR schemes

We begin by introducing the following JCR methods that will be compared with our proposed JoCRAT algorithm.

- *Femtocaching* [10]. This state-of-the-art algorithm starts with all the caches being empty. Then, it iteratively places the file to the cache that yields

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



the highest cache hits neglecting the bandwidth constraints, i.e., at each iteration it maximizes the function $\sum_{k\in\mathcal{K}} \mathbf{1}_{\{\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn}} x_n^{f(k)}\geq 1\}}$, where $\mathbf{1}_{\{.\}}$ is the indicator function. Each user request is routed to the nearest (with respect to the number of hops) cache having stored the requested file following a path with available bandwidth (if any).

- *Max Popularity*. Each cache $n$ stores independently the $C_n$ most popular files. Each user request is routed to the nearest cache having stored the requested file following a path with available bandwidth.
- *LP-ROUND* [29]. This randomized rounding algorithm solves the linear relaxation of JoCRAT and then rounds $\mathbf{x}$ and $\mathbf{y}$ into integers as follows. For each cache-node $n$, it sorts the files $f \in \mathcal{F}$ into decreasing order of the fractional values of the relaxed $x_{nf}, \forall f \in \mathcal{F}$, and greedily stores the top $C_n$ files at the cache $n$. Each user $k$ sorts routing paths $p \in \mathcal{P}_{kn}, \forall n \in \mathcal{N}_c$ into descending order with respect to the fractional values of the relaxed $y_{kn}^p, \forall n \in \mathcal{N}_c, \forall p \in \mathcal{P}_{kn}$, and routes the request of $k$ along the paths to the first accessible cache (if any).
- *Lag-Dual* [15]. This Lagrangian heuristic applies the hierarchical primal-dual decomposition method to approximate the optimal solution of JoCRAT iteratively. Specifically, at each iteration, each cache $n$ calculates the caching policy $\mathbf{x}$ based on the optimal dual variables of constraints in (6). Then, each link $l$ solves routing policy $\mathbf{y}$ based on the optimal dual variables of constraints in (4). The algorithm repeats a fixed number of iterations unless the stop criterion is met.
- *LR-JoCRAT*. The upper bound of the optimal solution value found by solving the linear relaxation of the JoCRAT problem.

It is well-known that greedy caching schemes, such as Femtocaching [10] and the variant in [16], perform nearly optimal if link bandwidth is fairly enough or there is only one congestible single-hop path. Meanwhile, the heuristics based on Lagrangian dual are of high efficiency in hierarchical networks [3], [12], [15]. We therefore need to ask whether these classic JCR schemes can reap the benefits of routing diversity when the links are congestible in the general topology.

### B. Evaluation setup

The main part of the evaluation is carried out for networks of Poisson geometric topology. Specifically, we consider a 1000 meters $\times$ 1000 meters geographical region where $N = 50$ nodes are deployed at random. 10% of the nodes in the network are equipped with caches. We say a pair of nodes are connected by a wireless link only when the distance between them is less than 200 meters. For each pair of nodes $k$ and $n$, there is a set $\mathcal{P}_{kn}$ that contains the three shortest (with respect to the number of hops) paths for routing the traffic between them. The bandwidth capacity of each link $l$ is set to $B_l = 45$ files during the time period. The storage capacity of each cache $n$ is set to $C_n = 100$ files.

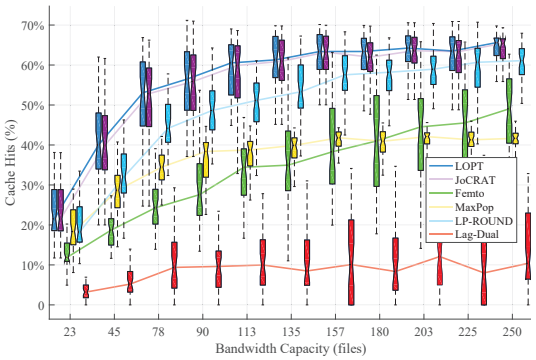


Fig. 4. Impact of bandwidth capacity.

Requests for $|\mathcal{F}| = 10,000$ files are generated by users that are associated with the nodes for a time period. The number of users is set to $|\mathcal{K}| = 2,250$, i.e., 50 users per node (except from the 5 caches). The popularity of file requests follows the Zipf distribution [37]. Namely, the probability of requesting the $i^{th}$ most popular file is proportional to $i^{-z}$ for some shape parameter $z > 0$ ($z = 0.9$).

We note that all the above values are varied during the evaluations. Besides, to smoothen the impact of randomness on the location of nodes (and hence topology) and file requests, the results we present are taken from the average of 40 evaluations. Confidence intervals are also depicted in the following figures to illustrate the statistic details of the simulations.

### C. Evaluation results

We first explore the impact of bandwidth capacity $B_l$, $\forall l$ on the cache hits. In Figure 4, $B_l$ spans a wide range of values, starting from 1 file to 250 files during time period. Supposed that the size of a file is of about 15 MB, and the time period lasts one minute. Then, the lower extreme bandwidth rates are of around 750 MB per minute or about 0.1 Gbps, which can be achieved by current LTE networks, while the upper extreme bandwidth rates are of 3750 MB per minute or about 0.5 Gbps, which is easier achieved by today's Internet Service Provider (*ISP*) backbone networks. Therefore, the evaluations capture a range of scenarios and types of networks, including the common IoT over the anticipated 5G networks. While the cache hits increase with $B_l$ for all the algorithms, the proposed JoCTAT algorithm performs close-to-optimal and better than the simplified randomized rounding LP-ROUND, Max Popularity, Femtocaching, and Lag-dual. The benefits are more pronounced for low and moderate bandwidth, while Femtocaching and LP-ROUND gradually approach JoCRAT as bandwidth increases. We note that Lag-Dual performs significantly worse than the other algorithms. This is because we let the Lag-Dual run for 100 iterations only, which takes time similar to the other algorithms (about 20 minutes), so as to make a fair comparison. Overall, *JoCRAT can even double the cache hits compared to its counterparts*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



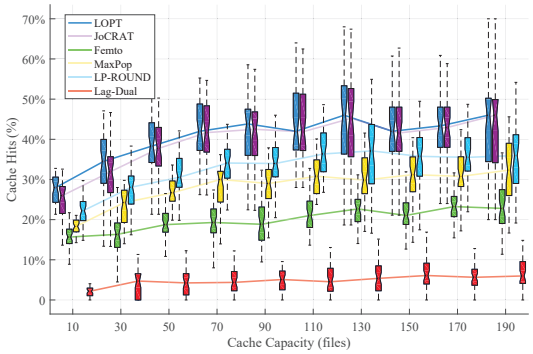


Fig. 5. Impact of cache capacity.

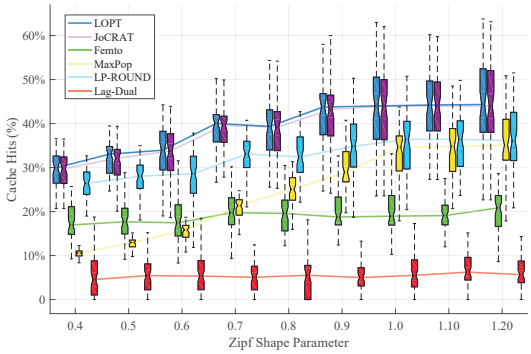


Fig. 6. Impact of Zipf shape parameter.

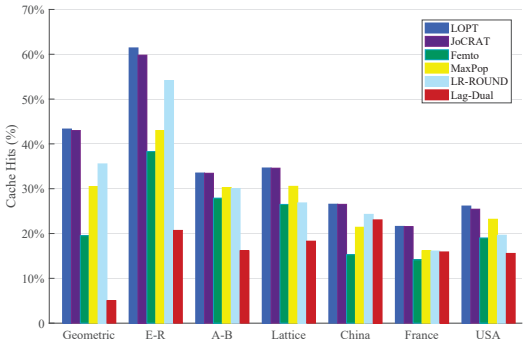


Fig. 7. Results for different networks.

Interestingly, Max Popularity performs better than Femtocaching in the low bandwidth regime, and vice versa in the high bandwidth regime. This can be attributed to the fact that Femtocaching tends to diversify the files among the caches, so that users can find more files at the accessed caches. However, such diversity is only useful when there is enough bandwidth for the users to access and download files from the caches. For low bandwidth, instead, duplicating the most popular files across caches can be more useful as there is not enough bandwidth to serve less popular files in the first place.

The impact of cache capacity $C_n$ is next studied in Figure 5. As expected, adding more cache capacities increases cache hits for all the algorithms, as more files are made available by the caches. The rate of growth gradually decreases, however, and the cache hits saturate eventually. The intuition is straightforward, namely, when the network is heavily congested, some end-users perhaps cannot access any of the cache-nodes and therefore many of file requests may not be served due to the lack of available bandwidth. The proposed JoCRAT algorithm and LP-ROUND perform consistently better than Max Popularity, Femtocaching and Lag-Dual. Interestingly, when bandwidth is limited, a simplified randomized rounding algorithm, i.e., LP-ROUND, also achieves relatively high cache hits, defeating the rest of JCR schemes except for the proposed JoCRAT. We address that our proposed JoCRAT always stays on the top no matter when bandwidth is limited or adequated. This implies that the core procedure in proposed JoCRAT, i.e., the risk evaluation, is indispensable in optimally utilizing the available storage and bandwidth resources.

In the following, we analyze the impact of the Zipf shape parameter $z$ on algorithms' performance as illustrated in Figure 6. On the one hand, as the $z$ value increases the file request distribution becomes steeper and a few files attract most of the requests. On the other hand, a small $z$ value corresponds to an almost uniform file request distribution. We observe that the cache hits increase with $z$ for LP-ROUND, and JoCRAT and Max Popularity algorithms. This is because the files cached by these algorithms attract more requests resulting in more cache hits. However, the cache hits remain roughly the same for Femtocaching as well as Lag-Dual. This can be attributed again to the fact that both Femtocaching and Lag-Dual tend to diversify the files among the caches. JoCRAT avoids this problem by strategically storing the popular files in a way that load balances the links and utilizes the available bandwidth the most.

Finally, we repeat the evaluations for different network topologies, both real and synthetic, as shown in Figure 7. Specifically, we use the real topologies of China Telecom (44 nodes, 62 links), France Renater (43 nodes, 56 links) and USA Deltatelecom (113 nodes, 183 links) that are available in [38]. We randomly deploy 4, 4, and 11 caches in these networks, respectively. We also generate synthetic topologies using the Erdős-Rényi (E-R) [39], Lattice, Geometric and Albert-Barabási (A-B) [40] randomized graph models (49 nodes for the Lattice, 50 nodes for the rest). Interestingly, the Lagrangian heuristic approaches the nearly-optimal in China Telecom network and France Renater network. This is reasonable since the topologies of these two networks are hierarchical and there is small duality gap under the layered topology. In contrast, our proposed JoCRAT is totally topology-free, i.e., we achieve nearly-optimal performance in almost all the networks regardless of the bandwidth and connectivity constraints. The results verify the superiority of the proposed JoCRAT algorithm over the state-of-the-art schemes.

## VII. Conclusion

Despite the recent spur of research in designing in-network caching algorithms, most of the previous works

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



were based on networks of special topologies (e.g., two-tier, hierarchical) or assuming non-congestible bandwidth capacities. Our work in this paper filled this important gap in the literature by considering the joint caching and routing problem in its general form. We formulated this problem for the objective of maximizing the number of content requests served by the caches (or cache hits) and analyzed its complexity. By using a randomized rounding technique, we proposed a solution algorithm and proved its approximation ratio. Evaluation results demonstrated the performance benefits of this algorithm over the state-of-the-art methods, as well as the key system parameters that affect these benefits.

## APPENDIX A
## PROOF OF LEMMA 3

The aggregate risk of violating the constraints or not reaching the number of cache hits specified by the approximation ratio can be written as $\mathbb{P}\{\bigcup_{l\in\mathcal{L}}\mathbf{E}_l\bigcup\bigcup_{n\in\mathcal{N}}\mathbf{E}_n\bigcup\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}$. This can be upper bounded by the potential energy function (PE) as follows:

$$\begin{aligned}
&\mathbb{P}\{\bigcup_{l\in\mathcal{L}}\mathbf{E}_l\bigcup\bigcup_{n\in\mathcal{N}}\mathbf{E}_n\bigcup\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}\\
&\overset{(a)}{\leq}\mathbb{P}\{\bigcup_{l\in\mathcal{L}}\mathbf{E}_l\bigcup\bigcup_{n\in\mathcal{N}}\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}+\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}\\
&\overset{(b)}{=}1-\mathbb{P}\{\bigcap_{l\in\mathcal{L}}\mathbf{E}_l^c\bigcap\bigcap_{n\in\mathcal{N}}\mathbf{E}_n^c|\mathbf{y}^*(\mathcal{U})\}+\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}\\
&\overset{(c)}{\leq}1+\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}\\
&-\prod_{l\in\mathcal{L}}(\mathbb{P}\{\mathbf{E}_l^c|\mathbf{y}^*(\mathcal{U})\})\times\prod_{n\in\mathcal{N}}(\mathbb{P}\{\mathbf{E}_n^c|\mathbf{y}^*(\mathcal{U})\})\\
&=1+\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\}\\
&-\prod_{l\in\mathcal{L}}(1-\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{U})\})\times\prod_{n\in\mathcal{N}}(1-\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{U})\}),
\end{aligned}\tag{27}$$

where we have denoted by $\mathbf{E}^c$ the complementary of event $\mathbf{E}$. Inequality (a) follows since the Boole's inequality. Equation (b) holds because of De Morgan's law. What remains to show is inequality (c). To this end, it suffices to show that:

$$\begin{aligned}
&\mathbb{P}\{\bigcap_{l\in\mathcal{L}}\mathbf{E}_l^c\bigcap\bigcap_{n\in\mathcal{N}}\mathbf{E}_n^c|\mathbf{y}^*(\mathcal{U})\}\\
&\geq\prod_{l\in\mathcal{L}}(\mathbb{P}\{\mathbf{E}_l^c|\mathbf{y}^*(\mathcal{U})\})\times\prod_{n\in\mathcal{N}}(\mathbb{P}\{\mathbf{E}_n^c|\mathbf{y}^*(\mathcal{U})\}
\end{aligned}\tag{28}$$

The above inequality can be directly proved by using the FKG inequality [41], a fundamental tool that tightly bounds the likelihood of concurrently happening a set of *positively correlated events*.

*Lemma 4: The FKG inequality* [41]: Define by $\mathbf{y}$ a random vector whose elements are independent but non-identical binary random variables. Consider two different outcomes of the random vector $\mathbf{y}'$ and $\mathbf{y}''$. We denote by $\mathbf{y}'\succeq\mathbf{y}''$ if and only if $\mathbf{y}'$ is element-wise larger than $\mathbf{y}''$. We refer to $\mathbf{E}$ as a *decreasing event* if $\mathbb{P}\{\mathbf{E}|\mathbf{y}'\}=1$ implies that $\mathbb{P}\{\mathbf{E}|\mathbf{y}''\}=1$, $\forall\mathbf{y}'\succeq\mathbf{y}''$. Let $\mathbf{E}_1,\mathbf{E}_2,\cdots,\mathbf{E}_i,\cdots$ be a sequence of decreasing events. Then, for any non-empty set $\mathcal{I}$, it holds that: $\mathbb{P}\{\bigcap_{i\in\mathcal{I}}\mathbf{E}_i\}\geq\prod_{i\in\mathcal{I}}\mathbb{P}\{\mathbf{E}_i\}$.

To apply the FKG inequality into (28), however, we need to show that the events $\mathbf{E}_l^c,\forall l\in\mathcal{L}$ and $\mathbf{E}_n^c,\forall n\in\mathcal{N}$ are decreasing. Specifically, we consider a current set of examined users $\mathcal{U}'\subseteq\mathcal{K}$ and a current rounded solution $\mathbf{y}^*(\mathcal{U}')$ during our algorithm execution such that $\mathbb{P}\{\mathbf{E}_l^c|\mathbf{y}^*(\mathcal{U}')\}=1$ for all links $l$. Then, we remove a subset of users $\mathcal{S}$ from $\mathcal{U}'$ to construct another set $\mathcal{U}''=\mathcal{U}'\setminus\mathcal{S}$, and update $\mathbf{y}^*(\mathcal{U}'')$ as follows:

$$y^{*p}_{kn}(\mathcal{U}'')=\begin{cases}y^{*p}_{kn}(\mathcal{U}'), & \text{if } k\notin\mathcal{S},\\ 0, & \text{if } k\in\mathcal{S}.\end{cases}$$

Obviously, we have $\mathbf{y}^*(\mathcal{U}')\succeq\mathbf{y}^*(\mathcal{U}'')$, and the term $B_l-\sum_{k\in\mathcal{U}}\big[\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn}:l\in p}y^{*p}_{kn}\big]$ (remaining bandwidth of link $l$) will not decrease if we replace $\mathcal{U}=\mathcal{U}'$ with $\mathcal{U}=\mathcal{U}''$. Therefore, by equation (14), $\mathbb{P}\{\mathbf{E}_l^c|\mathbf{y}(\mathcal{U}')\}=1$ implies that $\mathbb{P}\{\mathbf{E}_l^c|\mathbf{y}(\mathcal{U}'')\}=1$. This means that $\mathbf{E}_l^c$ is a decreasing event. Similarly, we can show the same property for $\mathbf{E}_n^c$ event, thus, conclude the proof of lemma 3.

## APPENDIX B
## PROOF OF THEOREM 3

We will prove Theorem 3 by showing that the PE value will be zero in the end of the JoCRAT algorithm execution. Therefore, the aggregate risk of violating the constraints or not reaching the approximation ratio will be zero.

We note that after the last iteration of the algorithm (when $\mathcal{U}=\mathcal{K}$), each of the risks $\mathbb{P}\{\mathbf{E}_l|\mathbf{y}^*(\mathcal{K})\}$, $\mathbb{P}\{\mathbf{E}_n|\mathbf{y}^*(\mathcal{K})\}$ and $\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{K})\}$ will be either zero or one. This is due to the expressions in equations (17), (20) and (23). Therefore, the PE value $h(\mathbf{y}^*(\mathcal{K}))$ will be zero, one or two (cf. equation (26)). To prove that it will be zero, and therefore Theorem 3 holds, we will show the following two claims:

1) we can always find a $\gamma$ value such that the initial PE $h(\mathbf{y}^*(\emptyset)))$ value is strictly less than 1, and
2) the PE $h(\mathbf{y}^*(\mathcal{U})$ values always decrease or keep the same value during the iteration over users thus the value can always keep being less than 1.

Therefore, the final PE $h(\mathbf{y}^*(\mathcal{K}))$ value must be zero. We formally prove the above claims below, in Propositions 1 and 2.

*Proposition 1:* There always exists a value $\gamma\in\big(e,\big(\frac{2Re^{M+1}}{e-1}\big)^{\frac{1}{M-2}}\big)$ such that the initial PE value $h(\mathbf{y}^*(\emptyset))$ is strictly less than 1.

To prove the above proposition, we will use the following concentration inequalities.

*Lemma 5 (The generic Chernoff bounds [42]):* Consider a set of independent Bernoulli random variables $y_1,y_2,\cdots,y_t,\cdots$ with respective success probabilities $\bar{y}_1,\bar{y}_2,\cdots,\bar{y}_t,\cdots$. Let $\mu\triangleq\mathbb{E}\big\{\sum_{t\in\mathcal{T}}y_t\big\}=\sum_{t\in\mathcal{T}}\bar{y}_t$. Then, it

holds that:

$$\mathbb{P}\left\{\sum_{t\in\mathcal{T}} y_t \geq \mu(1+\delta)\right\} \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}, \forall \delta > 1, \tag{29}$$

and,

$$\mathbb{P}\left\{\sum_{t\in\mathcal{T}} y_t \leq \mu(1-\delta)\right\} \leq e^{\frac{-\mu\delta^2}{2}}, \forall \delta \in (0,1), \tag{30}$$

In inequality (29), let $\delta > e-1$, $\gamma > \delta+1 > e$, $B_l = (1+\delta)\times\frac{\sum \bar{y}^p_{kn}}{\gamma} > \sum \bar{y}^p_{kn}$, and $\mu = \frac{\sum \bar{y}^p_{kn}}{\gamma}$. Then, $\mathbb{P}\{\mathbf{E}_l\} = \mathbb{P}\{\sum_{k\in\mathcal{K}}[\sum_{n\in\mathcal{N}_c}\sum_{p\in\mathcal{P}_{kn}:l\in p} y^p_{kn}] > B_l\}$ is strictly less than $\left(\frac{e}{\gamma}\right)^{B_l}$. The same goes for $1-\mathbb{P}\{\mathbf{E}_n\} \geq 1-\left(\frac{e}{\gamma}\right)^{C_n}$. However, in order to get a lower bound for $\mathbb{P}\{\mathbf{E}_{ch}\}$ some additional effort is needed. Specifically, we use the following lemma.

*Lemma 6 (Goemans-Williamson inequality [43]):* Consider a sequence $\bar{\mathbf{y}} = (\bar{y}_0, \cdots, \bar{y}_t, \cdots)$. Then, $\forall t \in \mathcal{T}$ and $\bar{y}_t \in [0,1]$, it holds that:

$$(1-\frac{1}{e})\min\left\{\sum_{t\in\mathcal{T}}\bar{y}_t, 1\right\} \leq 1-\prod_{t\in\mathcal{T}}(1-\bar{y}_t) \leq \min\left\{\sum_{t\in\mathcal{T}}\bar{y}_t, 1\right\}. \tag{31}$$

By using the above lemma, we can show that:

$$\begin{aligned} 1-\Pi_{p\in\mathcal{P}_{kn}}\Pi_{n\in\mathcal{N}}(1-\frac{\bar{y}^p_{k,n}}{\gamma}) &\geq (1-\frac{1}{e})\min\left\{\sum_{p\in\mathcal{P}_{kn}}\sum_{n\in\mathcal{N}}\frac{\bar{y}^p_{k,n}}{\gamma}, 1\right\} \\ &= (1-\frac{1}{e})\sum_{p\in\mathcal{P}_{kn}}\sum_{n\in\mathcal{N}}\frac{\bar{y}^p_{k,n}}{\gamma} \end{aligned} \tag{32}$$

and

$$\mu = \mathbb{E}\left\{\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}_{kn}} y^p_{k,n}\right\} = \frac{\text{LOPT}}{\gamma} \geq (1-\frac{1}{e})\times\frac{\text{LOPT}}{\gamma} \tag{33}$$

Due to (30), we have:

$$\begin{aligned} &\mathbb{P}\left\{\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}_{kn}} y^p_{k,n} \leq \frac{\text{LOPT}\times(1-\frac{1}{e})}{\gamma}\right\} \\ &\leq e^{\frac{-\text{LOPT}(1-\frac{1}{e})}{2}\times(1-\frac{1}{\gamma})^2} \leq 1-\frac{e-1}{2e}\times\frac{1}{\gamma^2} \end{aligned} \tag{34}$$

where $\gamma \geq e$ and we have set $\delta = 1-\frac{1}{\gamma}$.

To conclude the proof, we will use the bounds $1-\mathbb{P}\{\mathbf{E}_l\} \geq 1-\left(\frac{e}{\gamma}\right)^{B_l}$, $1-\mathbb{P}\{\mathbf{E}_n\} \geq 1-\left(\frac{e}{\gamma}\right)^{C_n}$, and $\mathbb{P}\{\mathbf{E}_{ch}\} \leq 1-\frac{e-1}{2e}\times\frac{1}{\gamma^2}$ shown above. We first define $M = \min_{l\in\mathcal{L},n\in\mathcal{N}}\{C_n, B_l\}$, $R = |\mathcal{N}_c|+|\mathcal{L}|$ and assume that $\gamma > e$. Then, we note that since $\forall x \in \mathbb{R}$, $x \geq -1$ it holds that $\forall n \in \mathbb{N}$: $(1+x)^n > 1+nx$, an easier sufficient condition to ensure that the initial PE value is strictly less than one is $1-\left(\frac{e}{\gamma}\right)^M R \geq 1-\frac{e-1}{2e\gamma^2}$. By substituting $\gamma = \left(\frac{2Re^{M+1}}{e-1}\right)^{\frac{1}{M-2}}$ into the above expression, we conclude the proof.

*Proposition 2:* Given the set of already examined users $\mathcal{U}$, there always exists a routing strategy $\{y^{p*}_{kn}, \forall n \in \mathcal{N}_c, p* \in \mathcal{P}_{kn} | \sum y^{p*}_{kn} \leq 1\}$ for the current user $k \notin \mathcal{U}$ such that the next PE value $h(\mathbf{y}^*(\mathcal{U}\cup\{k\}))$ is less or equal to the current PE value $h(\mathbf{y}^*(\mathcal{U}))$.

To facilitate the proof, we introduce the notation $\mathcal{I}_R \triangleq \{\mathbf{E}^c_l, \forall l \in \mathcal{L}\} \cup \{\mathbf{E}^c_n, \forall n \in \mathcal{N}\}$. Then, we refer to the product $\prod_{l\in\mathcal{L}}(\mathbb{P}\{\mathbf{E}^c_l|\mathbf{y}^*(\mathcal{U})\}) \times \prod_{n\in\mathcal{N}}(\mathbb{P}\{\mathbf{E}^c_n|\mathbf{y}^*(\mathcal{U})\})$ as $\prod_{i\leq R}\left(\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*(\mathcal{U})\}\right)$.

We also denote by $\mathbf{y}^*_{np}(\{k\})$ the case where we choose the routing strategy $\{y^{p*}_{kn}, \forall n \in \mathcal{N}_c, p* \in \mathcal{P}_{kn} | \sum y^{p*}_{kn} \leq 1\}$ for the current user $k$, i.e., to route the file request of the current user $k$ along the path $p$ towards cache node $n$. Similarly, by $\mathbf{y}^*_{\emptyset}(\{k\})$ we denote the case where we refuse to serve the request of user $k$ by the caches.

In order to prove that the PE value can always decrease or keep the same value, it suffices to show that there is at least one convex combination of all the possible values of $h(\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\}))$ in which $y^p_{kn} = 1, y^{p'}_{kn'} = 0 : \forall n \in \mathcal{N}_c, p \in \mathcal{P}_{kn}, p' \neq p$ is larger than the value of $h(\mathbf{y}^*(\mathcal{U}))$. We turn to prove the following easier sufficient condition:

$$\begin{aligned} &\prod_{i\leq R}\left(\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*(\mathcal{U})\}\right) - \mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\} \leq \\ &\sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}}\left[\bar{y}^p_{kn}\times\left(\prod_{i\leq R}\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\} - \mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\}\right)\right] + \left(1-\sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}}\bar{y}^p_{kn}\right) \\ &\times\left(\prod_{i\leq R}\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} - \mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\}\right), \end{aligned} \tag{35}$$

where the right hand side (R.H.S) expression is a convex combination of the PE values $h(\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})), \forall n \in \mathcal{N}_c, p \in \mathcal{P}_{kn}$.

We will prove that (35) holds for all $\mathcal{I}_R$ by using mathematical induction. Specifically, we define an $(|\mathcal{L}|+|\mathcal{N}_c|)$-size increasing sequence of the subsets of $\mathcal{I}_R$ by $\mathcal{I}_1 \subsetneq \mathcal{I}_2 \subsetneq \cdots, \mathcal{I}_r \subsetneq \cdots \subsetneq \mathcal{I}_R$, where $|\mathcal{I}_r| = r$.

**A. Case r = 1.** We show that for any index $1 \leq i \leq |\mathcal{I}_R|$ it holds that:

$$\begin{aligned} \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*(\mathcal{U})\} &\overset{a1}{=} \sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}}\left[\bar{y}^p_{kn}\times\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\}\right] \\ &+\left(1-\sum_{p\in\mathcal{P}_{kn}:n\in\mathcal{N}_c}\bar{y}^p_{kn}\right)\times\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\}, \end{aligned} \tag{36}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2935742, IEEE Internet of Things Journal



and,

$$\mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*(\mathcal{U})\} \overset{a2}{=} \sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \Bigg[\bar{y}^p_{kn} \times \mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\} \Bigg] + \Bigg(1 - \sum_{p\in\mathcal{P}_{kn}:n\in\mathcal{N}_c} \bar{y}^p_{kn}\Bigg) \times \mathbb{P}\{\mathbf{E}_{ch}|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} \tag{37}$$

where equations (a1) and (a2) are valid because of the definitions in (17), (20), and (23). Specifically, these equations hold because the $\mathbf{y}^*_{np}(\{k\}), \forall p \in \mathcal{P}_{kn}, \forall n \in \mathcal{N}_c$ are independently rounded to integers as in Algorithm 1, and the law of total probability. By substituting the above equations into the left hand side (*L.H.S.*) of (35), we show that it is valid for the case of $r = 1$.

**B. Case 2 $< \mathbf{r} <$ R.** We suppose that (35) is valid for $\mathcal{I}_r$ where $1 \leq r < R$. Substituting (37) into the L.H.S. of the case for $r$ , we can cancel the terms $\mathbb{P}\{\mathbf{E}_{ch}|\cdot\}$ in the both sides of (35) and obtain:

$$\prod_{i\leq r} \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*(\mathcal{U})\} \leq \sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \bar{y}^p_{kn} \times \kappa(r,k) + \Bigg(1 - \sum_{p\in\mathcal{P}_{kn}:n\in\mathcal{N}_c} \bar{y}^p_{kn}\Bigg) \times \nu(r,k), \tag{38}$$

where $\kappa(r,k) = \prod_{i\leq r} \mathbb{P}\big\{\mathbf{E}^c_i|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\big\}$ and $\nu(r,k) = \prod_{i\leq r} \big(\mathbb{P}\big\{\mathbf{E}^c_i|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\big\}$. We note that the product of the L.H.S. of (38) and $\mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*(\mathcal{U})\}$ is equivalent to $\prod_{i\leq r} \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*(\mathcal{U})\} \times \mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*(\mathcal{U})\}, \forall j > r$ for case $r+1$. We will complete the proof by showing that the product of the R.H.S. of (38) and the term $\mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*(\mathcal{U})\}$ is less or equal to the R.H.S. of the the sufficient condition in (35). To this end, we will prove by brute-force calculation that:

$$\sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \Bigg[\bar{y}^p_{kn} \times \kappa(r,k) \times \Bigg(\mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*(\mathcal{U})\}\Bigg)\Bigg] + \Bigg(1 - \sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \bar{y}^p_{kn}\Bigg) \times \nu(r,k) \times \Bigg(\mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*(\mathcal{U})\}\Bigg)$$

is at most:

$$\sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \Bigg[\bar{y}^p_{kn} \times \kappa(r+1,k)\Bigg] + \Bigg(1 - \sum_{\substack{p\in\mathcal{P}_{kn}\\ n\in\mathcal{N}_c}} \bar{y}^p_{kn}\Bigg) \times \nu(r+1,k),$$

The key is to substitute (36) into the above and then to cancel the following term:

$$\sum_{\substack{p,p'\in\mathcal{P}_{kn}\\ n,n'\in\mathcal{N}_c}} \bar{y}^p_{kn}\bar{y}^{p'}_{kn'} \Bigg[\kappa(r,k) + \nu(r,k)\Bigg] \times \Bigg[\mathbb{P}\{\mathbf{E}_j|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} - \mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*_{n'p'}(\mathcal{U}\cup\{k\})\}\Bigg].$$

Then, the residual is:

$$\Bigg(1 - \sum_{p\in\mathcal{P}_{kn}:n\in\mathcal{N}_c} \bar{y}^p_{kn}\Bigg) \times \sum_{p\in\mathcal{P}_{kn}:n\in\mathcal{N}_c} \bar{y}^p_{kn} \Bigg[ \prod_{i\leq r} \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} - \prod_{i\leq r} \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\}\Bigg] \times \Bigg[\mathbb{P}\{\mathbf{E}_j|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} - \mathbb{P}\{\mathbf{E}^c_j|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\}\Bigg] \geq 0. \tag{39}$$

Inequality (39) is valid by the fact that $\forall i \in \mathcal{I}_R, \forall k, \forall p \in \mathcal{P}_{kn} : n \in \mathcal{N}_c$, it holds that: $\mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{\emptyset}(\mathcal{U}\cup\{k\})\} - \mathbb{P}\{\mathbf{E}^c_i|\mathbf{y}^*_{np}(\mathcal{U}\cup\{k\})\} \geq 0, \forall 1 \leq r \leq R$. Intuitively, the risks of violating the bandwidth and cache capacities do not increase as the number of users served by the remote server increases. Therefore, inequality (35) is valid for $\mathcal{I}_{r+1}$.

## REFERENCES


[1] M. Z. Shafiq, A. X. Liu, and A. R. Khakpour, "Revisiting caching in content delivery networks", in *the Proc. ACM SIGMETRICS*, Austin, TX, USA, 2014.

[2] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2127–2145, Apr. 2018.

[3] X. Sun and N. Ansari, "Dynamic resource caching in the IoT application layer for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 606–613, Apr. 2018.

[4] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for Internet of Things: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2074–2083, Apr. 2019.

[5] Z. Lv, T. Yin, X. Zhang, H. Song, and G. Chen, "Virtual reality smart city based on WebVRGIS," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1015–1024, Dec. 2016.

[6] B. Liu, T. Jiang, Z. Wang, and Y. Cao, "Object-oriented network: A named-data architecture toward the future internet," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 957–967, Aug. 2017.

[7] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous internet of things build our future: A survey," *IEEE Commun. Surv. Tuts*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart. 2018.

[8] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems", *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411-1429, Aug. 2008.

[9] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010.

[10] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Sept. 2011.

[11] S. Ioannidis and E. Yeh, "Jointly optimal routing and caching for arbitrary network topologies," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1258–1275, June 2018.

[12] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution", in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012.

[13] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks", *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[14] A. Khreishah, J. Chakareski, and A. Gharaibeh, "Joint caching, routing, and channel assignment for collaborative small-cell cellular networks", *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2275–2284, Aug. 2016.

[15] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks", *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, Aug. 2016.

[16] M. Dehghan, B. Jiang, A. Seetharam, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal request routing and content caching in heterogeneous cache networks", *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1635–1648, June 2017.

[17] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing", Appear to *IEEE Trans. Mobile Comput.*, 2019.

[18] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing", *IEEE/ACM Trans. Netw.*, vol 26, no. 1, pp. 274–287, Feb. 2018.

[19] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "VIP: A framework for joint dynamic forwarding and caching in named data networks", in *Proc. ACM ICN*, Paris, France, 2014.

[20] J. Lee, M. Sviridenko, and J. Vondrak, "Submodular maximization over multiple matroids via generalized exchange properties", *Math. Oper. Res.*, vol. 35, no. 4, pp.795–806, Nov. 2010.

[21] A. Sadeghi, F. Sheikholeslami, and G. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities", *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[22] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution", *Comput. New.*, vol. 47, no. 3, pp. 409–428, Feb. 2005.

[23] G. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks", *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1111–1125, June 2018.

[24] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, , "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience", *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, Mar. 2017.

[25] A. Khreishah, H. B. Salameh, I. Khalil, and A. Gharaibeh, "Renewable energy-aware joint caching and routing for green communication networks", *IEEE Syst. J.*, vol. 12, no. 1, pp. 768–777, Mar. 2018.

[26] H. Hsu, and K. C. Chen, "A resource allocation perspective on caching to achieve low latency", *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 145–148, Jan. 2016.

[27] S. Sardellitti, F. Costanzo, and M. Merluzzi, "Joint optimization of caching and transport in proactive edge cloud", in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, 2018.

[28] M. Mahdian, E. Yeh, "MinDelay: Low-latency joint caching and forwarding for multi-hop networks", in *Proc. IEEE ICC*, Kansas City, MO, USA, 2018.

[29] T. He, H. Khamfroush, S. Wang, T. L. Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources", in *Proc. IEEE ICDCS*, Vienna, Austria, 2018.

[30] K. Poularakis, J. Llorca, A. Tulino, I. Taylor, L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks", in *Proc. IEEE Infocom*, 2019

[31] E. Bastug, J. L. Guenego, and M. Debbah, "Proactive small cell networks", in *Proc. IEEE ICT*, Casablanca, Morocco, 2013.

[32] 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.

[33] O. Azar, and Yossiand Regev, "Combinatorial algorithms for the unsplittable flow problem," *Algorithmica*, vol. 44, no. 1, pp. 49–66, Jan. 2006.

[34] L. Fleischer, M. X. Goemans, V. Mirrokni, and M. Sviridenko, "Tight approximation algorithms for maximum general assignment problems," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Miami, FL, USA, 2006.

[35] Y. Hong, "On computing the distribution function for the Poisson binomial distribution," *Comput. Stat. Data Anal.*, vol. 59, no. 1, pp. 41–51, Mar. 2013.

[36] A. Srinivasan, "Approximation algorithms via randomized rounding: A survey," in *Series in Advanced Topics in Mathematics*. Polish Scientific Publishers, 1999.

[37] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems", *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1447–1460, Dec. 2008.

[38] S. Knight, H. X. Hung, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011.

[39] P. Erdős and R. Alfréd, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, Jan. 1960.

[40] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–511, Oct. 1999.

[41] R. L. Graham, "Applications of the FKG inequality and its relatives," [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-68874-46

[42] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 1–13, Mar. 1963.

[43] M. X. Goemans and D. P. Williamson, "New $\frac{3}{4}$-approximation algorithms for the maximum satisfiability problem," *SIAM J. Discret. Math.*, vol. 7, no. 4, pp. 656–666, Nov. 1994.