

Stochastic Content-Centric Multicast Scheduling for Cache-Enabled Heterogeneous Cellular Networks

Bo Zhou, Ying Cui, *Member, IEEE*, and Meixia Tao, *Senior Member, IEEE*

Abstract—Caching at small base stations (SBSs) has demonstrated significant benefits in alleviating the backhaul requirement in heterogeneous cellular networks (HetNets). While many existing works focus on what contents to cache at each SBS, an equally important problem is what contents to deliver so as to satisfy dynamic user demands given the cache status. In this paper, we study optimal content delivery in cache-enabled HetNets by taking into account the inherent multicast capability of wireless medium. We consider stochastic content multicast scheduling to jointly minimize the average network delay and power costs under a multiple access constraint. We establish a content-centric request queue model and formulate this stochastic optimization problem as an infinite horizon average cost Markov decision process (MDP). By using *relative value iteration* and special properties of the request queue dynamics, we characterize some properties of the value function of the MDP. Based on these properties, we show that the optimal multicast scheduling policy is of threshold type. Then, we propose a structure-aware optimal algorithm to obtain the optimal policy. We also propose a low-complexity suboptimal policy, which possesses similar structural properties to the optimal policy, and develop a low-complexity algorithm to obtain this policy.

Index Terms—Heterogeneous cellular networks, wireless caching, content-centric, multicast, Markov decision process, structural properties, queueing.

I. INTRODUCTION

The rapid proliferation of smart mobile devices has triggered an unprecedented growth of the global mobile data traffic, which is expected to reach 24.3 exabytes per month by 2019 [2]. One promising approach to meet the dramatic traffic growth is to deploy small base stations (SBSs) together with traditional macro base stations (MBSs) in a heterogeneous network paradigm [3]. Such a heterogeneous cellular network (HetNet) provides short-range localized communications by bringing base stations (BSs) closer to users, and hence increases the area spectral efficiency and network capacity. However, the main drawback of this approach is the requirement of expensive high-speed backhaul links for connecting all SBSs to the core network. The backhaul capacity requirement can be enormously high during peak traffic hours.

Recently, caching at BSs has been proposed as an effective way to alleviate the backhaul capacity requirement and improve the user-perceived quality of experience in wireless networks [4]–[7]. Caching has received significant attention in the

literature [8]–[12]. Specifically, in [8], the authors introduce the concept of *FemtoCaching* and study content placement at SBSs to minimize the average content access delay. In [9], the authors consider joint request routing and content caching for HetNets and propose approximate algorithms to minimize the content access delay. In [10], the authors study the joint optimization of power and cache control for video streaming in multi-cell multi-user MIMO systems. References [11] and [12] study outage probability and average delivery rate of cache-enabled HetNets for given caching strategies. However, [8]–[12] consider point-to-point unicast transmission for cache-enabled wireless networks and can only help to reduce the backhaul burden without effectively relieving the “on air” congestion. Furthermore, in [8]–[12], the inherent broadcast nature of wireless medium is not fully exploited, which is the major distinction of wireless communications from wired communications.

Enabling multicasting at BSs is another efficient approach to deliver popular contents to multiple requesters concurrently. Wireless multicasting has been specified in 3GPP standards known as evolved Multimedia Broadcast Multicast Service (eMBMS) [13]. In view of the benefits of caching and multicasting, joint design of the two promising techniques is expected to achieve superior performance for massive content delivery in wireless networks. From an information-theoretic perspective, [14] proposes a novel coded caching scheme to minimize the peak traffic load for a single-cell network by utilizing multicast transmission and caches at users, and characterizes the memory-rate tradeoff. Note that, [14] only considers the delay-insensitive services. However, many content-centric applications, such as video steaming, are delay-sensitive, and it is critical to consider delay performance in cache-enabled content-centric wireless networks [15]–[18]. In specific, the authors in [15] study coded multicasting for inelastic services (with strict deadline) in a single-cell network, and propose a computationally efficient content delivery algorithm under a given coded caching scheme. In [16], the authors propose an approximate caching algorithm with performance guarantee and a heuristic caching algorithm, to reduce the service cost for inelastic services in a cache-enabled small-cell network under a given multicast transmission strategy. In [17], the authors consider multicasting for inelastic services in a cache-enabled multi-cell network, and propose joint throughput-optimal caching and scheduling algorithms to maximize the service rates of the inelastic services. In our recent work [18], we consider optimal multicast scheduling to jointly minimize the average delay and service costs of elastic services (delay-sensitive services but without strict deadlines) for a cache-

This paper was presented in part at ACM CoNEXT 2015 Workshop on Content Caching and Delivery in Wireless Networks (CCDWN), Heidelberg, Germany, December, 2015 [1].

B. Zhou, Y. Cui and M. Tao are with the Department of Electronic Engineering at Shanghai Jiao Tong University, Shanghai, 200240, P. R. China. Email: {b.zhou, cuiying, mxtao}@sjtu.edu.cn.

enabled single-cell network. However, for elastic services, it remains unknown how to design optimal multicast scheduling to jointly minimize average delay and service costs for given cache placement in cache-enabled HetNets. The main challenge of determining optimal multicast scheduling in HetNets stems from the heterogeneous structure of the network (see a motivating example in Section II-D).

In this paper, we consider a cache-enabled HetNet with one MBS, N SBSs, K users, and M contents (with possibly different content sizes). The SBS coverage areas are assumed to be disjoint. Assume that the MBS and the SBSs are not allowed to operate concurrently, to avoid excessive interference, while the SBSs are allowed to operate at the same time without mutual interference. This is referred to as the multiple access constraint. Each SBS is equipped with a cache storing a certain number of contents, depending on the sizes of the cached contents and the cache size. The MBS stores all contents in the network. In each slot, each BS either schedules one cached content for multicasting to serve the pending requests from the users in its coverage area, or keeps idle, i.e., does not transmit any content. We consider stochastic content multicast scheduling to jointly minimize the average network delay and power costs under the multiple access constraint. We establish a content-centric request queue model and then formulate this stochastic multicast scheduling problem as an infinite horizon average cost Markov decision process (MDP) [19]. There are several technical challenges involved.

- **Optimality analysis:** The infinite horizon average cost MDP is well-known to be challenging due to the curse of dimensionality. Although dynamic programming provides a systematic approach for MDPs, there generally exist only numerical solutions. These solutions do not typically offer many design insights and are usually impractical due to the curse of dimensionality [19]. Thus, it is highly desirable to study the structural properties of the optimal policy. There are several existing works on structural analysis for hybrid systems [20]–[22]. However, the system models and queueing models in these works significantly differ from ours, and hence the approaches and results therein cannot be straightforwardly extended to our work. Specifically, our problem can be viewed as a problem of scheduling a single broadcast server (the MBS) or multiple multicast servers (the SBSs) to parallel (request) queues with general arrivals. Existing works have only studied the problems of scheduling a single broadcast server to parallel queues [18], [23], [24]. Therefore, the structural analysis of the optimal multicast scheduling of a single broadcast server or multiple broadcast servers to parallel queues with general arrivals remains unknown and cannot be straightforwardly extended from the existing solutions.

- **Algorithm design:** Standard numerical algorithms such as value iteration and policy iteration to MDPs are usually computationally impractical for real systems. To reduce the complexity, several structured optimal algorithms, which incorporate the structural properties into standard algorithms, are proposed [25], [26]. However, the curse of dimensionality still remains an issue. On the other hand, the structural properties of the optimal policy may be one key reason for its good performance. Thus, it is highly desirable to further reduce

the complexity of the structured optimal algorithms, while maintaining similar structural properties to the optimal policy.

By using *relative value iteration algorithm* (RVIA) [19, Chapter 4.3.1] and special properties of the request queue dynamics, we characterize some properties of the value function of the MDP. Based on these properties, we show that the optimal multicast scheduling policy, which is adaptive to the request queue state, is of threshold type. This reveals the tradeoff between the average delay cost and the average power cost. Then, we propose a structure-aware optimal algorithm by exploiting the structural properties of the optimal policy. To further reduce the computational complexity, using approximate dynamic programming [19], we propose a low-complexity suboptimal policy, which possesses similar structural properties to the optimal policy, and develop a low-complexity algorithm to compute this policy. Numerical examples verify the theoretical results and demonstrate the performance of the optimal and suboptimal solutions.

The rest of this paper is organized as follows. Section II introduces the network model. Section III provides the formulation of the stochastic multicast scheduling problem and the optimality equation. The structural properties of the optimal policy are presented in Section IV and a structure-aware optimal algorithm is proposed in Section V. In Section VI, we propose a low-complexity suboptimal solution. Numerical results are provided in Section VII. Finally, we conclude the paper and provide several future directions in Section VIII. The important notations used in this paper are summarized in Table I.

TABLE I: List of important notations

\mathcal{N}^+	set of all SBSs
$\mathcal{N} = \mathcal{N}^+ \cup \{0\}$	set of all BSs; BS 0: MBS
\mathcal{K}_0	set of users not covered by any SBS
$\mathcal{K}_n, n \in \mathcal{N}^+$	set of users within coverage area of SBS n
$\mathcal{K} = \bigcup_{n \in \mathcal{N}^+} \mathcal{K}_n$	set of all users
\mathcal{M}	set of all contents
$\mathcal{M}_n, n \in \mathcal{N}$	set of contents cached at BS n
\mathcal{N}_m	set of SBSs that caching content m
n, k, m, t	BS, user, content, slot index
$p(n, m)$	minimum transmission power required by BS n to deliver content m to all associated users within a slot
$\mathbf{A} = (A_{n,m})$	request arrival matrix
$\mathbf{Q} = (Q_{n,m})$	request queue state matrix
$d(\mathbf{Q})$	sum request queue length
$\mathbf{u} = (u_n)_{n \in \mathcal{N}}$	multicast scheduling action
μ	stationary multicast scheduling policy
$V(\mathbf{Q})$	value function
$J(\mathbf{Q}, u)$	state-action cost function

II. NETWORK MODEL

Consider a cache-enabled HetNet with one MBS, N SBSs, K users, and M contents, as illustrated in Fig. 1. Let $\mathcal{N} \triangleq \{0, 1, 2, \dots, N\}$ denote the set of all BSs, where BS 0 refers to the MBS and BS $n = 1, 2, \dots, N$ refers to SBS n . Let $\mathcal{N}^+ \triangleq \{1, 2, \dots, N\}$ denote the set of N SBSs. The SBS coverage areas are assumed to be disjoint. Let $\mathcal{K} \triangleq \{1, 2, \dots, K\}$ denote the set of K users in the network. Let $\mathcal{K}_n \subseteq \mathcal{K}$ denote the set of users within the coverage area of SBS $n \in \mathcal{N}^+$. All the users in \mathcal{K}_n ($n \in \mathcal{N}^+$) can be

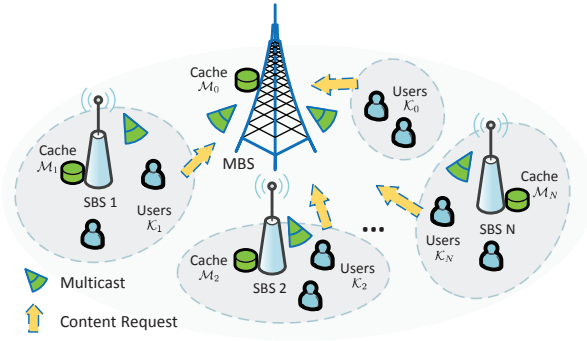


Fig. 1: Cache-enabled heterogeneous cellular network.

served by the MBS and SBS n . Let $\mathcal{K}_0 \triangleq \mathcal{K} - \bigcup_{n \in \mathcal{N}^+} \mathcal{K}_n$ denote the set of users not covered by any SBS. All the users in \mathcal{K}_0 can only be served by the MBS. Note that, we do not distinguish the users in \mathcal{K}_n . Let $\mathcal{M} \triangleq \{1, 2, \dots, M\}$ denote the set of M contents (with possibly different content sizes) in the network. Each BS is equipped with a cache storing a certain number of contents, depending on the cache size and the sizes of the cached contents. Let $\mathcal{M}_n \subseteq \mathcal{M}$ denote the set of cached contents in BS $n \in \mathcal{N}$. We assume $\mathcal{M}_0 = \mathcal{M}$, i.e., the MBS stores all contents in the network. Let $\mathcal{N}_m \triangleq \{n \in \mathcal{N}^+ | m \in \mathcal{M}_n\}$ denote the set of SBSs caching content $m \in \mathcal{M}$. We assume that the contents stored in the caches are given according to certain caching strategy (similar assumptions have been made in the literature, e.g. [11], [12], [15]) and consider multicast scheduling for a given caching design. Notice that caching in general is in a much larger time-scale (e.g., on a weekly or monthly basis) while multicast scheduling is in a shorter time-scale [4], [8], [10]. Consider time slots of unit length (without loss of generality) indexed by $t = 1, 2, \dots$. In the sequel, we first introduce the request arrival model, the service model, and the request queue model. Then, we provide a motivating example that highlights the challenges in designing the optimal multicast scheduling.

A. Request Arrival Traffic

In each slot, each user submits content requests to the MBS. Let $A_{n,m}(t) \in \{0, 1, \dots\}$ denote the number of the requests for content m from all users in \mathcal{K}_n which arrive during slot t , where $n \in \mathcal{N}$ and $m \in \mathcal{M}$. Let $\mathbf{A}(t) \triangleq (A_{n,m}(t))_{n \in \mathcal{N}, m \in \mathcal{M}}$ denote the request arrival matrix during slot t . We assume that the request arrival processes $\{A_{n,m}(t)\}$ ($n \in \mathcal{N}, m \in \mathcal{M}$) are mutually independent with respect to n and m ; and $\{A_{n,m}(t)\}$ are i.i.d. with respect to t for all $n \in \mathcal{N}$ and $m \in \mathcal{M}$. Note that, the content request arrivals are modeled according to the Independent Reference Model (IRM), which is a standard approach adopted in the literature [8], [17], [27]. The MBS maintains separate request queues for each BS $n \in \mathcal{N}$ and each cached content $m \in \mathcal{M}_n$. The request queue model will be further illustrated in Section II-C.

B. Service Model

We consider multicast service for content delivery in the network. For ease of illustration, we assume that each BS

can transmit at most one content in each slot. The analytical framework and results can be extended to the general case in which each BS can transmit multiple contents in one slot. In each slot, each BS $n \in \mathcal{N}$ either schedules one cached content for multicasting to serve the pending requests from all users in its coverage area, or keeps idle (i.e., does not transmit any content). We consider the Zero Download Delay (ZDD) assumption [27], i.e., all scheduled contents in one slot can be delivered to the users within the same slot. Let $p(n, m)$ denote the minimum transmission power required by BS n for successfully delivering one cached content m to all users in its coverage area within a scheduling slot, where $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. We set $p(n, 0) = 0$ for all $n \in \mathcal{N}$. If BS n multicasts content m with transmission power $p(n, m)$, all pending requests for content m from all users in the coverage area of BS n are satisfied. Let $u_n(t) \in \tilde{\mathcal{M}}_n \triangleq \mathcal{M}_n \cup \{0\}$ denote the scheduling action of BS $n \in \mathcal{N}$ at slot t , where $u_n(t) \neq 0$ indicates that BS n multicasts the cached content $u_n(t)$ with transmission power $p(n, u_n(t))$ at slot t , and $u_n(t) = 0$ indicates that BS n does not transmit any content at slot t . Let $\mathbf{u}(t) \triangleq (u_n(t))_{n \in \mathcal{N}}$ denote the multicast scheduling action in the network at slot t .

The MBS is assumed to operate at a much higher transmission power level than the SBSs, for providing full coverage of the network. To avoid excessive interference, we therefore do not allow the MBS and the SBSs to operate concurrently. On the other hand, since the SBSs are spatially separated and use much lower powers, we allow the SBSs to operate at the same timewithout mutual interference. Mathematically, for all t , we require,

$$u_0(t) \sum_{n \in \mathcal{N}^+} u_n(t) = 0. \quad (1)$$

We refer to (1) as the multiple access constraint. Let $\mathcal{U} \triangleq \{(u_n)_{n \in \mathcal{N}} | u_n \in \tilde{\mathcal{M}}_n \forall n \in \mathcal{N} \text{ and } u_0 \sum_{n \in \mathcal{N}^+} u_n = 0\}$ denote the feasible multicast scheduling action space. The network power cost $p(\mathbf{u})$ associated with $\mathbf{u} \in \mathcal{U}$ is given by

$$p(\mathbf{u}) \triangleq \sum_{n \in \mathcal{N}} p(n, u_n). \quad (2)$$

C. Request Queue Model

As illustrated above, for each SBS $n \in \mathcal{N}^+$, the requests for cached content $m \in \mathcal{M}_n$ from all the users in \mathcal{K}_n can be served by both the MBS and SBS n , while the requests for uncached content $m \in \mathcal{M}_0 \setminus \mathcal{M}_n$ can only be served by the MBS. On the other hand, the MBS can serve the requests for any content $m \in \mathcal{M}_0$ from the users in \mathcal{K}_n . Therefore, the request queues maintained by the MBS are constructed as follows. For each SBS $n \in \mathcal{N}^+$ and each cached content $m \in \mathcal{M}_n$, we construct a separate request queue, referred to as queue (n, m) , storing the requests for content m from all the users in \mathcal{K}_n . Let $Q_{n,m}(t)$ denote the length of queue (n, m) at the beginning of slot t , where $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. For the MBS $n = 0$ and each content $m \in \mathcal{M}_0$, we also construct a separate request queue, referred to as queue $(0, m)$, storing the requests for content m from all users in $\bigcup_{n \in \mathcal{N}^+ \setminus \mathcal{N}_m} \mathcal{K}_n$ (the set of users covered by the SBSs where content m is

not cached) and \mathcal{K}_0 . Let $Q_{0,m}(t)$ denote the length of queue $(0,m)$ at the beginning of slot t , where $m \in \mathcal{M}_0$. Note that these request queues can be implemented using counters and no data is contained in these queues. For each queue (n,m) , let $N_{n,m}$ denote the upper limit of the corresponding counter. For technical tractability, we assume that $N_{n,m}$ is finite (can be arbitrarily large). This assumption is to guarantee that the request queue state space is finite, which would greatly simplify the mathematical arguments [19]. Let $\mathcal{Q}_{n,m} \triangleq \{0, 1, \dots, N_{n,m}\}$ denote the request queue state space for queue (n,m) . Let $\mathbf{Q}(t) \triangleq (Q_{n,m}(t))_{n \in \mathcal{N}, m \in \mathcal{M}_n} \in \mathcal{Q}$ denote the request queue state of the network at the beginning of slot t , where $\mathcal{Q} \triangleq \prod_{n \in \mathcal{N}} \prod_{m \in \mathcal{M}_n} \mathcal{Q}_{n,m}$ denotes the request queue state space and the \prod operation denotes the Cartesian product.

For each SBS $n \in \mathcal{N}^+$ and each cached content $m \in \mathcal{M}_n$, all the pending requests in queue (n,m) are satisfied, if content m is scheduled for multicasting by the MBS (i.e., $u_0(t) = m$) or by SBS n (i.e., $u_n(t) = m$) at slot t . Thus, for each $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$, the request queue dynamics is as follows:

$$Q_{n,m}(t+1) = \min\{\mathbf{1}(u_0(t) \neq m \ \& \ u_n(t) \neq m)Q_{n,m}(t) + A_{n,m}(t), N_{n,m}\}, \quad (3)$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

For the MBS and each content $m \in \mathcal{M}_0$, all the pending requests in queue $(0,m)$ are satisfied, if content m is scheduled for multicasting by the MBS at slot t (i.e., $u_0(t) = m$). Thus, for the MBS and each content $m \in \mathcal{M}_0$, the request queue dynamics is as follows:

$$Q_{0,m}(t+1) = \min\{\mathbf{1}(u_0(t) \neq m)Q_{0,m}(t) + \tilde{A}_{0,m}(t), N_{0,m}\}, \quad (4)$$

where $\tilde{A}_{0,m}(t) \triangleq \sum_{n \in \mathcal{N}^+ \setminus \mathcal{N}_m} A_{n,m}(t) + A_{0,m}(t)$ denotes the total number of requests for content m from all users in $\bigcup_{n \in \mathcal{N}^+ \setminus \mathcal{N}_m} \mathcal{K}_n$ and \mathcal{K}_0 which arrive during slot t . Note that each request is stored in only one queue.

D. Motivating Example

As illustrated in Fig. 2, consider a network with 1 MBS, 2 SBSs ($\mathcal{N}^+ = \{1, 2\}$), 3 users ($\mathcal{K} = \{1, 2, 3\}$), and 3 contents ($\mathcal{M} = \{1, 2, 3\}$). We set $\mathcal{K}_1 = \{1\}$, $\mathcal{K}_2 = \{2\}$, $\mathcal{K}_0 = \{3\}$, $\mathcal{M}_1 = \{1, 2\}$, $\mathcal{M}_2 = \{2, 3\}$, and $\mathcal{M}_0 = \{1, 2, 3\}$. According to Section II-C, the MBS maintains seven request queues, i.e., queues $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 1)$, $(1, 2)$, $(2, 2)$, and $(2, 3)$. Our goal is to design the optimal multicast scheduling so as to jointly minimize the average network delay cost and power cost. This involves two challenging and coupled tasks. First, at each time slot, it is not clear whether to operate the MBS or the SBSs. If we schedule the MBS to multicast, then the pending requests for one content in the whole network can be satisfied with a higher power cost, e.g., clear queues $(0, 2)$, $(1, 2)$, and $(2, 2)$ with power $p(0, 2)$. If we schedule the SBSs to multicast, the pending requests for (possibly different) contents in different SBS coverage areas can be satisfied with a lower power cost, e.g., clear queues $(1, 2)$ and $(2, 3)$ with power $p(1, 2) + p(2, 3)$.

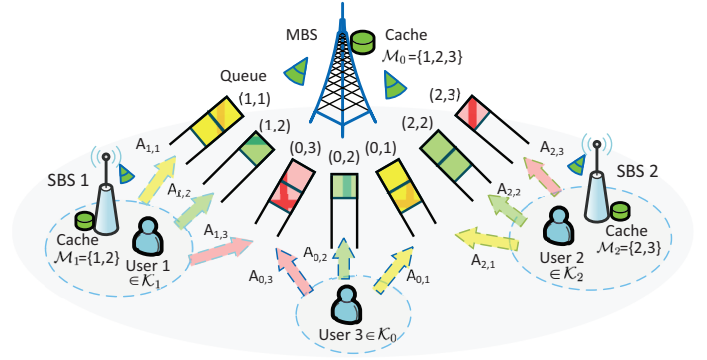


Fig. 2: An example with 1 MBS, 2 SBSs, 3 users and 3 contents.

Second, at each time slot, if BS $n \in \mathcal{N}$ is allowed to operate, it is unknown whether to keep BS n idle or not, and which content to schedule for multicasting if BS n is not idle. Take SBS 1 as an example. Suppose at certain time slot t we have $Q_{1,1}(t) > Q_{1,2}(t)$ and $p(1, 1) > p(1, 2)$. Scheduling Content 1 can satisfy more requests with a higher power cost; scheduling Content 2 can satisfy fewer requests with a lower power cost; keeping SBS 1 idle consumes zero power cost.

From this example, we can see that, the challenges in designing the optimal multicast scheduling come from the heterogeneous structure of the network, and the difficulty in balancing the delay cost and the power cost. In the sequel, we formalize the multicast scheduling problem and try to tackle these challenges.

III. PROBLEM FORMULATION AND OPTIMALITY EQUATION

A. Problem Formulation

Given an observed request queue state \mathbf{Q} , the multicast scheduling action \mathbf{u} is determined according to a stationary policy defined below.

Definition 1 (Stationary Policy): A feasible stationary multicast scheduling policy μ is a mapping from the request queue state $\mathbf{Q} \in \mathcal{Q}$ to the feasible multicast scheduling action $\mathbf{u} \in \mathcal{U}$, where $\mu(\mathbf{Q}) = \mathbf{u}$.

By the queue dynamics in (3) and (4), the induced random process $\{\mathbf{Q}(t)\}$ under policy μ is a controlled Markov chain with the following transition probability:

$$\begin{aligned} \Pr[\mathbf{Q}'|\mathbf{Q}, \mathbf{u}] &\triangleq \Pr[\mathbf{Q}(t+1) = \mathbf{Q}' | \mathbf{Q}(t) = \mathbf{Q}, \mathbf{u}(t) = \mathbf{u}] \\ &= \mathbb{E}[\Pr[\mathbf{Q}(t+1) = \mathbf{Q}' | \mathbf{Q}(t) = \mathbf{Q}, \mathbf{u}(t) = \mathbf{u}, \mathbf{A}(t) = \mathbf{A}]], \end{aligned} \quad (5)$$

where the expectation is taken over the distribution of the request arrival \mathbf{A} and

$$\begin{aligned} \Pr[\mathbf{Q}(t+1) = \mathbf{Q}' | \mathbf{Q}(t) = \mathbf{Q}, \mathbf{u}(t) = \mathbf{u}, \mathbf{A}(t) = \mathbf{A}] \\ = \begin{cases} 1, & \text{if } \mathbf{Q}' \text{ satisfies (3) and (4)} \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

We restrict our attention to stationary unichain policies.¹ For a given stationary unichain policy μ , the average network

¹A unichain policy is a policy, under which the induced Markov chain has a single recurrent class (and possibly some transient states) [19].

delay cost is defined as

$$\bar{d}(\mu) \triangleq \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[d(\mathbf{Q}(t))], \quad (6)$$

where $d(\mathbf{Q}) \triangleq \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} Q_{n,m}$ and the expectation is taken with respect to the measure induced by the random request arrivals and the policy μ . According to Little's law, $\bar{d}(\mu)$ reflects the average waiting time in the network under policy μ . For a given stationary unichain policy μ , the average network power cost is given by

$$\bar{p}(\mu) \triangleq \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[p(\mathbf{u}(t))]. \quad (7)$$

In this paper, we would like to jointly minimize the average network delay cost and the average network power cost. We adopt the weighted-sum method, which is a commonly used method for multiobjective optimization problems [28]. Specifically, we define the average network cost as the weighted sum of the average network delay cost and the average network power cost, i.e.,

$$\bar{g}(\mu) \triangleq \bar{d}(\mu) + w\bar{p}(\mu) = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[g(\mathbf{Q}(t), \mathbf{u}(t))], \quad (8)$$

where $g(\mathbf{Q}, \mathbf{u}) \triangleq d(\mathbf{Q}) + wp(\mathbf{u})$ is the per-stage network cost and w is the weight indicating the relative importance of the average power cost over the average delay cost. Note that, w can also be treated as the penalty factor, mimicking the soft average delay constraint. In other words, w can be thought of a Lagrange multiplier on the average delay cost constraint [29].

We wish to find an optimal multicast scheduling policy to minimize the average network cost $\bar{g}(\mu)$ in (8).

Problem 1 (Network Cost Minimization):

$$\bar{g}^* \triangleq \min_{\mu} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[g(\mathbf{Q}(t), \mathbf{u}(t))], \quad (9)$$

where μ is a stationary unichain policy in Definition 1 and \bar{g}^* denotes the minimum average network cost achieved by the optimal policy μ^* .²

Notice that, if $w = 0$, Problem 1 reduces to the delay minimization problem, which can still be covered by our framework. However, if the power minimization is the sole goal, this optimization problem is no longer meaningful, as all BSs would keep idle and no requests would be served. In Problem 1, we assume the existence of a stationary unichain policy achieving the minimum in (9). Later in Lemma 1, we shall prove the existence of such a policy. Problem 1 is an infinite horizon average cost MDP, which is challenging due to the curse of dimensionality.

²Restricting the search for the optimal policy to unichain policies is to guarantee the existence of stationary optimal policies. This is widely used in the literature, e.g., [10], [25], [30].

B. Optimality Equation

The optimal multicast scheduling policy μ^* can be obtained by solving the following Bellman equation.

Lemma 1 (Bellman Equation): There exist a scalar θ and a value function $V(\cdot)$ satisfying:

$$\theta + V(\mathbf{Q}) = \min_{\mathbf{u} \in \mathcal{U}} \{g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V(\mathbf{Q}')] \}, \quad \forall \mathbf{Q} \in \mathcal{Q}, \quad (10)$$

where the expectation is taken over the distribution of the request arrival \mathbf{A} , and $\mathbf{Q}' \triangleq (Q'_{n,m})_{n \in \mathcal{N}, m \in \mathcal{M}_n}$ with $Q'_{0,m} \triangleq \min\{\mathbf{1}(u_0 \neq m)Q_{0,m} + \tilde{A}_{0,m}, N_{0,m}\}$ for all $m \in \mathcal{M}_0$ and $Q'_{n,m} \triangleq \min\{\mathbf{1}(u_0 \neq m \ \& \ u_n \neq m)Q_{n,m} + A_{n,m}, N_{n,m}\}$ for all $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. Furthermore, $\theta = \bar{g}^*$ is the optimal value to Problem 1 for all initial state $\mathbf{Q}(1) \in \mathcal{Q}$ and the optimal policy achieving the optimal value \bar{g}^* is given by

$$\mu^*(\mathbf{Q}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \{g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V(\mathbf{Q}')] \}, \quad \forall \mathbf{Q} \in \mathcal{Q}. \quad (11)$$

Proof: Please see Appendix A. ■

From Lemma 1, we observe that μ^* given by (11) depends on \mathbf{Q} through the value function $V(\cdot)$. Obtaining $V(\cdot)$ involves solving the Bellman equation in (10) for all $\mathbf{Q} \in \mathcal{Q}$, which does not admit a closed-form solution in general [19]. Standard numerical solutions such as value iteration and policy iteration are usually computationally impractical to implement, and do not typically yield many design insights [19]. Thus, it is highly desirable to study the structural properties of the optimal policy μ^* .

IV. OPTIMALITY PROPERTIES

Problem 1 can be viewed as a problem of scheduling a single broadcast server (the MBS) or multiple multicast servers (the SBSs) to parallel (request) queues with general arrivals. The structural analysis is more challenging than the existing structural analysis for the scheduling of a single broadcast server. First, by RVIA and the special structure of the request queue dynamics, we can prove the following property of the value function.

Lemma 2 (Monotonicity of $V(\mathbf{Q})$): For any $\mathbf{Q}^1, \mathbf{Q}^2 \in \mathcal{Q}$ such that $\mathbf{Q}^2 \succeq \mathbf{Q}^1$, we have $V(\mathbf{Q}^2) \geq V(\mathbf{Q}^1)$.³

Proof: Please see Appendix B. ■

Next, we introduce the state-action cost function:

$$J(\mathbf{Q}, \mathbf{u}) \triangleq g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V(\mathbf{Q}')]. \quad (12)$$

Note that $J(\mathbf{Q}, \mathbf{u})$ is related to the R.H.S. of the Bellman equation in (10). Then, based on $J(\mathbf{Q}, \mathbf{u})$, we introduce:

$$\Delta_{\mathbf{u}, \mathbf{v}}(\mathbf{Q}) \triangleq J(\mathbf{Q}, \mathbf{u}) - J(\mathbf{Q}, \mathbf{v}). \quad (13)$$

Note that $\Delta_{\mathbf{u}, \mathbf{v}}(\mathbf{Q}) = -\Delta_{\mathbf{v}, \mathbf{u}}(\mathbf{Q})$. Action \mathbf{u} is said to dominate \mathbf{v} at state \mathbf{Q} if $\Delta_{\mathbf{u}, \mathbf{v}}(\mathbf{Q}) \leq 0$. In particular, by Lemma 1, if \mathbf{u} dominates all $\mathbf{v} \in \mathcal{U}$ at state \mathbf{Q} , then $\mu^*(\mathbf{Q}) = \mathbf{u}$. Based on Lemma 2, we have the following property of the function defined in (13).

Lemma 3 (Monotonicity of $\Delta_{\mathbf{u}, \mathbf{v}}(\mathbf{Q})$): For any $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{U}$, $\Delta_{\mathbf{u}, \mathbf{v}}(\mathbf{Q})$ has the following properties.

³The notation \succeq indicates component-wise \geq .

- 1) If $u_0 \in \mathcal{M}_0$, then $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ is monotonically non-increasing with $Q_{0,m}$ and $Q_{n,m}$ for all $n \in \mathcal{N}_m$, where $m = u_0$.
- 2) If $u_n \in \mathcal{M}_n$ for some $n \in \mathcal{N}^+$, then $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ is monotonically non-increasing with $Q_{n,m}$, where $m = u_n$.

Proof: Please see Appendix C. \blacksquare

Lemma 3 indicates that, if \mathbf{u} dominates \mathbf{v} at some state \mathbf{Q} , then by increasing Q_{0,u_0} or Q_{n,u_0} for any $n \in \mathcal{N}_{u_0}$ and $u_0 \neq 0$, or by increasing Q_{n,u_n} for any $n \in \mathcal{N}^+$ and $u_n \neq 0$, \mathbf{u} still dominates \mathbf{v} . The properties of $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ in Lemma 3 are similar to the diminishing-return property of submodular functions used in the existing structural analysis [31]. Lemma 3 stems from the special properties of multicasting and is essential to characterize the optimality properties. By Lemma 3, we can characterize the structural properties of the optimal policy μ^* . We start with several definitions. Define

$$\begin{aligned} \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) &\triangleq \{Q_{n,m} | Q_{n,m} \in \mathcal{Q}_{n,m} \text{ and} \\ \Delta_{\mathbf{u},\mathbf{v}}(Q_{n,m}, \mathbf{Q}_{-n,-m}) \leq 0 \ \forall \mathbf{v} \in \mathcal{U} \text{ and } \mathbf{v} \neq \mathbf{u}\}, \end{aligned} \quad (14)$$

where $\mathbf{Q}_{-n,-m} \triangleq (Q_{i,j})_{i \in \mathcal{N}, j \in \mathcal{M}_i, (i,j) \neq (n,m)}$. Based on $\Phi_{\mathbf{u}}(\cdot)$, we define:

$$\phi_{\mathbf{u}}^+(\mathbf{Q}_{-n,-m}) \triangleq \begin{cases} \max \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}), & \text{if } \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \neq \emptyset \\ -\infty, & \text{otherwise} \end{cases}, \quad (15)$$

$$\phi_{\mathbf{u}}^-(\mathbf{Q}_{-n,-m}) \triangleq \begin{cases} \min \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}), & \text{if } \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \neq \emptyset \\ +\infty, & \text{otherwise} \end{cases}. \quad (16)$$

Let $\mathbf{0}$ denote the $1 \times (N+1)$ vector with all entries 0. Then, we have the following theorem.

Theorem 1 (Structural properties of μ^):* For any $\mathbf{Q} \in \mathcal{Q}$, the optimal policy μ^* has the following structural properties.

- 1) $\mu^*(\mathbf{Q}) = \mathbf{0}$ for all $\mathbf{Q} \in \mathcal{Q}_0 \triangleq \{\mathbf{Q} | Q_{n,m} \leq \phi_{\mathbf{0}}^+(\mathbf{Q}_{-n,-m}), \forall m \in \mathcal{M}_n \text{ and } n \in \mathcal{N}\}$.
- 2) If $\exists n \in \mathcal{N}$, such that $u_n^* \in \mathcal{M}_n$, then $\mu^*(\mathbf{Q}) = \mathbf{u}^*$ for all $\mathbf{Q} \in \mathcal{Q}$ such that

$$Q_{n,m} \geq \phi_{\mathbf{u}^*}^-(\mathbf{Q}_{-n,-m}), \quad (17)$$

where $m = u_n^*$. Moreover, $\phi_{\mathbf{u}^*}^-(\mathbf{Q}_{-0,-m})$ is monotonically non-increasing with $Q_{n,m}$ for all $n \in \mathcal{N}_m$.

Proof: Please see Appendix D. \blacksquare

We illustrate the analytical results of Theorem 1 in Fig. 3, where the optimal policy is computed numerically using *policy iteration algorithm* (PIA) [32, Chapter 8.6]. We observe from Fig. 3(a) that, if the queue state falls in the region of blue squares (i.e., \mathcal{Q}_0), the optimal control is $(0, 0)$, i.e., both the MBS and the SBS keep idle. Hence, we refer to \mathcal{Q}_0 as the idle region of the optimal policy. From Fig. 3(b)-3(d), we observe that given $\mathbf{Q}_{-n,-m}$, the scheduling for content $m \in \mathcal{M}_n$ by BS $n \in \mathcal{N}$ is of threshold type (Property 2 of Theorem 1). This indicates that, it is not efficient to schedule content m by BS n when $Q_{n,m}$ is small (i.e., the delay cost is small), as a higher power cost per request is consumed. This shows the tradeoff between the delay cost and the power cost. Fig. 3(c) illustrates the monotonically non-increasing property of $\phi_{\mathbf{u}^*}^-(\mathbf{Q}_{-0,-1})$ in terms of $Q_{1,1}$. This reveals that the MBS is more willing to

Algorithm 1 Structured Policy Iteration Algorithm

- 1: Set $\mu_0^*(\mathbf{Q}) = \mathbf{0}$ for all $\mathbf{Q} \in \mathcal{Q}$, select reference state \mathbf{Q}^\dagger , and set $l = 0$.
- 2: (Policy Evaluation) Given policy μ_l^* , compute the average cost θ_l and value function $V_l(\mathbf{Q})$ from the linear system of equations⁴

$$\begin{cases} \theta_l + V_l(\mathbf{Q}) = g(\mathbf{Q}, \mu_l^*(\mathbf{Q})) + \mathbb{E}[V_l(\mathbf{Q}')], \ \forall \mathbf{Q} \in \mathcal{Q} \\ V_l(\mathbf{Q}^\dagger) = 0 \end{cases}, \quad (20)$$

where \mathbf{Q}' is defined in Lemma 1.

- 3: (Structured Policy Improvement) Obtain a new policy μ_{l+1}^* , where for each $\mathbf{Q} \in \mathcal{Q}$, $\mu_{l+1}^*(\mathbf{Q})$ is such that: **if** $\exists n \in \mathcal{N}$ and $\mathbf{Q}' \in \mathcal{Q}$ such that $\mu_l^*(\mathbf{Q}') = \mathbf{u}$, $u_n \in \mathcal{M}_n$, $Q'_{n,m} \leq Q_{n,m}$, and $Q'_{i,j} = Q_{i,j}$ for all $(i,j) \neq (n,m)$, where $m = u_n$ **then**

$$\mu_{l+1}^*(\mathbf{Q}) = \mathbf{u}.$$

else

$$\mu_{l+1}^*(\mathbf{Q}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \{g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V_l(\mathbf{Q}')]\}.$$

endif

- 4: Go to Step 2 until $\mu_{l+1}^* = \mu_l^*$.
-

multicast content 1 when $Q_{1,1}$ is large. The reason is that the MBS can satisfy more requests than any SBS. These optimality properties provide design insights for multicast scheduling in practical cache-enabled HetNets.

V. STRUCTURE-AWARE OPTIMAL ALGORITHM

The results in Theorem 1 can be exploited to substantially reduce the computational complexity for solving the Bellman equation in (10) in obtaining μ^* . In particular, by Property 2 in Theorem 1, for all $\mathbf{Q} \in \mathcal{Q}$, $\mathbf{u} \in \mathcal{U}$ and $\mathbf{Q}' = (Q'_{n,m})_{n \in \mathcal{N}, m \in \mathcal{M}_n}$ satisfying

$$\begin{cases} Q'_{n,m} \geq Q_{n,m}, & \text{if } u_n = m \\ Q'_{n,m} = Q_{n,m}, & \text{if } u_n \neq m \end{cases} \quad (18)$$

for each $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$, we have

$$\mu^*(\mathbf{Q}) = \mathbf{u} \Rightarrow \mu^*(\mathbf{Q}') = \mathbf{u}. \quad (19)$$

Therefore, by incorporating the property in (19) into the standard PIA, we develop a structure-aware algorithm in Algorithm 1, which is referred to as the structured policy iteration algorithm (SPIA). According to Theorem 8.6.6 and Chapter 8.11.2 in [32], we know that SPIA converges to the optimal policy μ^* in (11) within a finite number of iterations, and hence is an optimal algorithm.

Note that, in Step 3 (structured policy improvement) of Algorithm 1, we do not need to perform the minimization over \mathcal{U} when the condition is satisfied (which is the case for a large amount of queue states in \mathcal{Q}). This can be seen in Fig. 3.

⁴The solution to (20) can be obtained directly using Gaussian elimination or iteratively using the relative value iteration method [19].

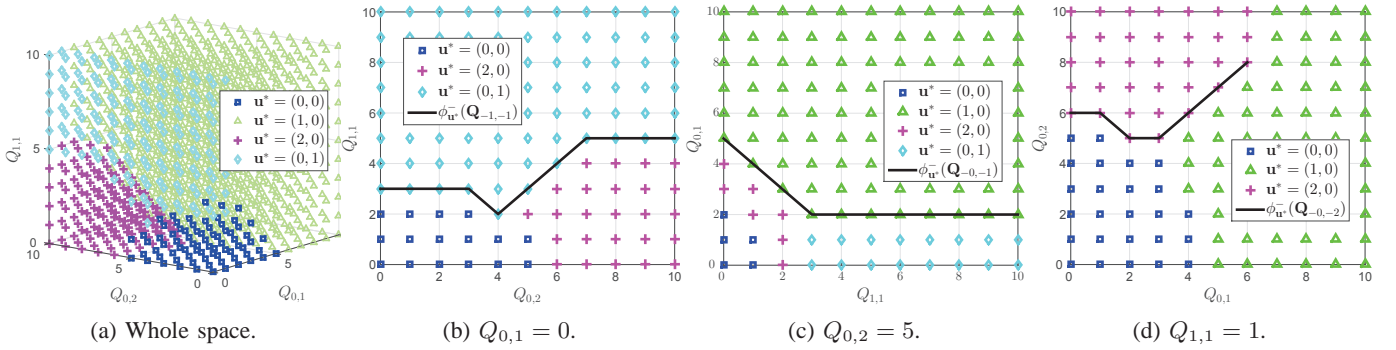


Fig. 3: Structure of the optimal multicast scheduling. $N = 1$, $\mathcal{K} = \{1, 2, 3, 4\}$, $\mathcal{M} = \{1, 2\}$, $\mathcal{K}_1 = \{1, 2\}$, and $\mathcal{M}_1 = \{1\}$. In each slot, each user requests one content, which is content 1 with probability 0.6 and content 2 with probability 0.4.

While, in the standard policy improvement step of PIA, the new policy μ_{l+1}^* is obtained by:

$$\mu_{l+1}^*(\mathbf{Q}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \{g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V_l(\mathbf{Q}')]\}, \quad \forall \mathbf{Q} \in \mathcal{Q}. \quad (21)$$

By (21), obtaining μ_{l+1}^* requires a brute-force minimization over \mathcal{U} for each $\mathbf{Q} \in \mathcal{Q}$, which can be very computationally expensive when the numbers of the contents M and the SBSs N are large. By comparing the structured policy improvement step of SPIA with the standard policy improvement step of PIA, we can see that SPIA can achieve considerable computational saving. Note that, the complexity of SPIA depends on the specific algorithm used for solving the linear system equations in (20). For instance, using Gaussian elimination for solving (20), the complexity of each iteration in SPIA is $O(|\mathcal{Q}|^3 + |\mathcal{U}||\mathcal{Q}|^2)$ [32].

Although the proposed structure-aware optimal algorithm, i.e., SPIA, can alleviate the computational burden of the standard PIA, it still suffers from the curse of dimensionality [19], due to the exponential growth of the cardinality of the system state space ($|\mathbf{Q}| = \prod_{n \in \mathcal{N}} \prod_{m \in \mathcal{M}_n} |Q_{n,m}|$). When the number of the SBSs and the cache sizes are large, the resulting huge state space may render SPIA computationally impractical.

VI. LOW COMPLEXITY SUBOPTIMAL SOLUTION

To further reduce the complexity of the proposed structure-aware optimal algorithm (i.e., SPIA) and relieve the curse of dimensionality, we would like to develop low-complexity suboptimal solutions. Note that the structural properties of the optimal policy may be one key reason that leads to good performance. Thus, in this section, we focus on design of suboptimal solutions which can maintain the structures of the optimal policy. Specifically, based on a randomized base policy, we first propose a low-complexity suboptimal deterministic policy using approximate dynamic programming [19]. We show that the deterministic policy improves the randomized base policy and possesses similar properties to the optimal policy. Then, we design a low-complexity structured algorithm to compute the proposed policy, by exploiting these structural properties.

A. Low Complexity Suboptimal Policy

The structural properties of the optimal policy come from the monotonicity property of the value function. Therefore, to maintain these structures in designing a suboptimal solution, we consider a value function decomposition method that can preserve the structural properties of the value function. Based on this decomposition, we propose a low-complexity suboptimal deterministic policy, which will be shown to possess similar structural properties to the optimal policy. We first introduce a randomized base policy.

Definition 2 (Randomized Base Policy): A randomized base policy for the multicast scheduling control $\hat{\mu}$ is given by a distribution on the feasible multicast scheduling action space \mathcal{U} .

We restrict our attention to randomized unichain base policies. Let $\hat{\theta}$ and $\hat{V}(\mathbf{Q})$ denote the average cost and the value function under a randomized unichain base policy $\hat{\mu}$, respectively. By [19, Proposition 4.2.2], there exists $(\hat{\theta}, \{\hat{V}(\mathbf{Q})\})$ satisfying:

$$\hat{\theta} + \hat{V}(\mathbf{Q}) = \mathbb{E}^{\hat{\mu}} [g(\mathbf{Q}, \mathbf{u})] + \sum_{\mathbf{Q}' \in \mathcal{Q}} \mathbb{E}^{\hat{\mu}} [\Pr[\mathbf{Q}'|\mathbf{Q}, \mathbf{u}]] \hat{V}(\mathbf{Q}'), \quad \forall \mathbf{Q} \in \mathcal{Q}, \quad (22)$$

Next, we show that $\hat{V}(\mathbf{Q})$ has an additive separable structure in the following lemma.

Lemma 4 (Additive Separable Structure of $\hat{V}(\mathbf{Q})$): Given any randomized unichain base policy $\hat{\mu}$, the value function $\{\hat{V}(\mathbf{Q})\}$ in (22) can be expressed as $\hat{V}(\mathbf{Q}) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \hat{V}_{n,m}(Q_{n,m})$, where $\{\hat{V}_{n,m}(Q_{n,m})\}$ satisfies:

$$\begin{aligned} \hat{\theta}_{n,m} + \hat{V}_{n,m}(Q_{n,m}) &= \mathbb{E}^{\hat{\mu}} [g_{n,m}(Q_{n,m}, \mathbf{u})] \\ &+ \sum_{Q'_{n,m} \in \mathcal{Q}_{n,m}} \mathbb{E}^{\hat{\mu}} [\Pr[Q'_{n,m}|Q_{n,m}, \mathbf{u}]] \hat{V}_{n,m}(Q'_{n,m}), \end{aligned} \quad \forall Q_{n,m} \in \mathcal{Q}_{n,m}, \quad (23)$$

for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. Here, $\hat{\theta}_{n,m}$ and $\hat{V}_{n,m}(Q_{n,m})$ denote the per-BS-content average cost and value function under $\hat{\mu}$, respectively, $g_{n,m}(Q_{n,m}, \mathbf{u}) \triangleq Q_{n,m} + w\mathbf{1}(u_n = m)p(n, m)$, and $\Pr[Q'_{n,m}|Q_{n,m}, \mathbf{u}] \triangleq \Pr[Q_{n,m}(t+1) = Q'_{n,m}|Q_{n,m}(t) = Q_{n,m}, \mathbf{u}(t) = \mathbf{u}]$.

Proof: Please see Appendix E. ■

To alleviate the curse of dimensionality, we approximate the value function in (10) with $\hat{V}(\mathbf{Q})$:

$$V(\mathbf{Q}) \approx \hat{V}(\mathbf{Q}) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \hat{V}_{n,m}(Q_{n,m}), \quad (24)$$

where $\{\hat{V}_{n,m}(Q_{n,m})\}$ is given by the per-BS-content fixed point equation in (23). Then, we develop the following low-complexity deterministic policy $\hat{\mu}^*$:

$$\hat{\mu}^*(\mathbf{Q}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \left\{ g(\mathbf{Q}, \mathbf{u}) + \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \mathbb{E} \left[\hat{V}_{n,m}(Q_{n,m}) \right] \right\}, \quad \forall \mathbf{Q} \in \mathcal{Q}. \quad (25)$$

In the following proposition, we show that the deterministic policy $\hat{\mu}^*$ generated by (25) always improves the corresponding randomized unichain base policy $\hat{\mu}$.

Proposition 1 (Performance Improvement): $\hat{\theta}^* \geq \hat{\theta}$, where $\hat{\theta}$ is the average network cost under a randomized unichain base policy $\hat{\mu}$ and $\hat{\theta}^*$ is the average network cost under the proposed solution.

Proof: This result follows directly from [33]. ■

Note that, to obtain $\hat{\mu}^*$ in (25) via solving (23) for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$, we only need to compute $\{\hat{V}_{n,m}(Q_{n,m})\}$ (a total of $O(\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} |Q_{n,m}|)$ values). The computational complexity is much lower than computing $\{V(\mathbf{Q})\}$ (a total of $O(\prod_{n \in \mathcal{N}} \prod_{m \in \mathcal{M}_n} |Q_{n,m}|)$ values) via solving (10) in obtaining μ^* in (11).

B. Structural Properties of Suboptimal Policy

In this part, we investigate the structural properties of the suboptimal policy $\hat{\mu}^*$ in (25). Along the lines of the structural analysis for the optimal policy in Section IV, we first introduce the state-action cost function for $\hat{\mu}^*$:

$$\hat{J}(\mathbf{Q}, \mathbf{u}) \triangleq g(\mathbf{Q}, \mathbf{u}) + \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \mathbb{E} \left[\hat{V}_{n,m}(Q_{n,m}) \right]. \quad (26)$$

Note that $\hat{J}(\mathbf{Q}, \mathbf{u})$ is related to the R.H.S. of (25). Then, we introduce:

$$\hat{\Delta}_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \triangleq \hat{J}(\mathbf{Q}, \mathbf{u}) - \hat{J}(\mathbf{Q}, \mathbf{v}), \quad (27)$$

$$\hat{\Phi}_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \triangleq \{Q_{n,m} | Q_{n,m} \in \mathcal{Q}_{n,m} \text{ and } \hat{\Delta}_{\mathbf{u},\mathbf{v}}(Q_{n,m}, \mathbf{Q}_{-n,-m}) \leq 0 \ \forall \mathbf{v} \in \mathcal{U} \text{ and } \mathbf{v} \neq \mathbf{u}\}. \quad (28)$$

By replacing $\Phi(\cdot)$ with $\hat{\Phi}(\cdot)$ in (15) and (16), we have $\hat{\phi}_{\mathbf{u}}^+(\cdot)$ and $\hat{\phi}_{\mathbf{u}}^-(\cdot)$, respectively. In the following theorem, we show that the proposed deterministic low-complexity suboptimal policy possesses similar structural properties to the optimal policy. This similarity would be one key reason for the good performance of the proposed suboptimal policy, as will be shown in Section VII.

Theorem 2 (Structural properties of $\hat{\mu}^$):* For any $\mathbf{Q} \in \mathcal{Q}$, the optimal policy $\hat{\mu}^*$ has the following structural properties.

- 1) $\hat{\mu}^*(\mathbf{Q}) = \mathbf{0}$ for all $\mathbf{Q} \in \hat{\mathcal{Q}}_0 \triangleq \{\mathbf{Q} | Q_{n,m} \leq \hat{\phi}_0^+(\mathbf{Q}_{-n,-m}), \forall m \in \mathcal{M}_n \text{ and } n \in \mathcal{N}\}$.
- 2) If $\exists n \in \mathcal{N}$, such that $\hat{u}_n^* \in \mathcal{M}_n$, then $\hat{\mu}^*(\mathbf{Q}) = \hat{\mathbf{u}}^*$ for all $\mathbf{Q} \in \mathcal{Q}$ such that

$$Q_{n,m} \geq \hat{\phi}_{\hat{\mathbf{u}}^*}^-(\mathbf{Q}_{-n,-m}). \quad (29)$$

where $m = \hat{u}_n^*$. Moreover, $\hat{\phi}_{\hat{\mathbf{u}}^*}^-(\mathbf{Q}_{-0,-m})$ is monotonically non-increasing with $Q_{n,m}$ for all $n \in \mathcal{N}_m$.

Proof: Please see Appendix F. ■

C. Structured Suboptimal Algorithm

By employing the relation between $\hat{\mu}$ and $\hat{\mu}^*$ as well as the structural properties of $\hat{\mu}^*$ in Theorem 2, we develop a low-complexity algorithm, referred to as the structured suboptimal algorithm (SSA), to obtain $\hat{\mu}^*$ in (25), as summarized in Algorithm 2. For SSA, it requires only one iteration to obtain $\hat{\mu}^*$. The complexity depends on the specific algorithm used for solving (23). For instance, using Gaussian elimination for solving (23), the complexity of SSA is $O(\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} |Q_{n,m}|^3 + |\mathcal{U}||\mathcal{Q}|^2)$.

Algorithm 2 Structured Suboptimal Algorithm

- 1: Given a randomized base unichain policy $\hat{\mu}$, compute the per-BS-content value function $\{\hat{V}_{n,m}(Q_{n,m})\}$ for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$ by solving the linear system of equations in (23).⁵
- 2: Obtain the deterministic policy $\hat{\mu}^*$, where for each $\mathbf{Q} \in \mathcal{Q}$, $\hat{\mu}^*(\mathbf{Q})$ is such that:
if $\exists n \in \mathcal{N}$ and $\mathbf{Q}' \in \mathcal{Q}$ such that $\hat{\mu}^*(\mathbf{Q}') = \mathbf{u}$, $u_n \in \mathcal{M}_n$, $Q'_{n,m} \leq Q_{n,m}$, and $Q'_{i,j} = Q_{i,j}$ for all $(i,j) \neq (n,m)$, where $m = u_n$ **then**

$$\hat{\mu}^*(\mathbf{Q}) = \mathbf{u}.$$

else

Compute $\hat{\mu}^*(\mathbf{Q})$ using (25).

endif

Now, we compare the computational complexity of SSA and SPIA. SSA is similar to one iteration of SPIA. As discussed in Section VI-A, in Step 1 of SSA, the number of value functions required to be computed is much smaller than that in each iteration of the policy evaluation step (Step 3 in Algorithm 1) of SPIA. In Step 2 of SSA, the number of optimizations required to solve is comparable to that in each iteration of the structured policy improvement step of SPIA. Therefore, SSA has a significantly lower computational complexity than SPIA.

VII. NUMERICAL RESULTS AND DISCUSSIONS

In this section, we evaluate the performance of the proposed optimal and suboptimal solutions through numerical examples. In the simulations, we assume that in each slot, each user requests one content (independently), which is content m with probability P_m . We assume that $\{P_m\}$ follows a (normalized) Zipf distribution with parameter α [34]. For simplicity, we assume that each content is of the same size and each SBS has the same cache size. These assumptions are commonly used in the literature on wireless caching in HetNets, e.g., [8]–[12]. As the coverage areas of the SBSs are assumed to be disjoint,

⁵The solution to (23) can be obtained directly using Gaussian elimination or iteratively using the relative value iteration method [19].

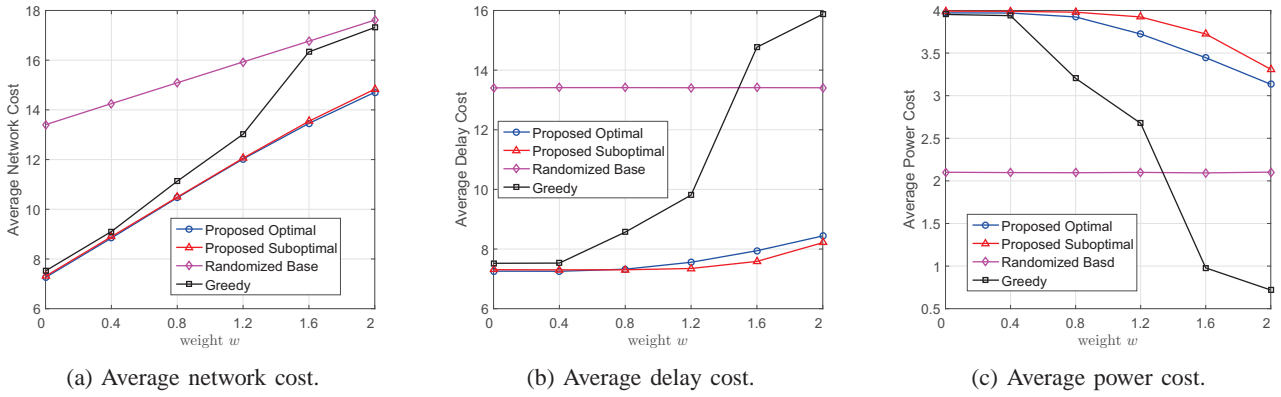


Fig. 4: Average network, delay, and power costs versus weight w .

we adopt a commonly used content placement strategy, i.e., each SBS stores the most popular contents [8], [11], [12].

First, we compare the average costs of the proposed optimal and suboptimal solutions with two baseline policies, i.e., a randomized base policy in Definition 2 and a greedy policy. In particular, for the randomized base policy, in each slot, we randomly select the MBS or all the SBSs to operate, with probability P_{MBS} and $1 - P_{MBS}$, respectively. Then, if BS n is allowed to operate, it keeps idle with probability P_{idle}^n or schedules one cached content $m \in \mathcal{M}_n$ for multicasting with probability $P_m^n = (1 - P_{idle}^n)P_m / \sum_{m \in \mathcal{M}_n} P_m$. Note that, the suboptimal policy is obtained based on this randomized base policy, as illustrated in Section VI. In the following simulations, we set $P_{MBS} = 0.5$ and $P_{idle}^n = 0.3$ for all $n \in \mathcal{N}$. To illustrate the greedy policy, we first introduce its cost function $C(\mathbf{Q}, \mathbf{u}) \triangleq \sum_{n \in \mathcal{N}} c_n(\mathbf{Q}, u_n)$, where $\mathbf{Q} \in \mathcal{Q}$, $\mathbf{u} \in \mathcal{U}$, and

$$c_0(\mathbf{Q}, u_0) \triangleq \begin{cases} wp(0, m) - Q_{0,m} - \sum_{n \in \mathcal{N}_m} Q_{n,m} & \text{if } u_0 = m \in \mathcal{M}_0, \\ 0 & \text{if } u_0 = 0 \end{cases},$$

$$c_n(\mathbf{Q}, u_n) \triangleq \begin{cases} wp(n, m) - Q_{n,m} & \text{if } u_n = m \in \mathcal{M}_n, \forall n \in \mathcal{N}^+, \\ 0 & \text{if } u_n = 0 \end{cases}.$$

For the greedy policy, in each slot, we choose the multi-cast scheduling action that minimizes the cost function, i.e., $\mathbf{u}(t) = \arg \min_{\mathbf{u}(t) \in \mathcal{U}} C(\mathbf{Q}(t), \mathbf{u}(t))$. This greedy policy can also be treated as an approximate solution to Problem 1 through approximating $V(\mathbf{Q})$ with \mathbf{Q} in (11). Note that this policy determines the scheduling action myopically, without accurately considering the impact of the action on the future costs. This type of policy is a commonly used baseline policy in the literature (see e.g., [35] and references therein).

Fig. 4 illustrates the average network cost, delay cost, and power cost versus the weight of the power cost (i.e., w) for a network with 1 MBS, 1 SBS, 4 users, and 3 contents ($\mathcal{M} = \{1, 2, 3\}$). We set $\alpha = 0.75$, $|\mathcal{K}_0| = |\mathcal{K}_1| = 2$, $\mathcal{M}_0 = \{1, 2, 3\}$, $\mathcal{M}_1 = \{1\}$, $p(0, m) = 4$ for all $m \in \mathcal{M}_0$, and $p(1, m) = 2$ for all $m \in \mathcal{M}_1$. It can be observed in Fig 4(a) that the average network costs of the proposed optimal and suboptimal policies are very close to each other, and are lower than those of the

randomized base policy and the greedy policy. The reason is that the proposed two policies can make foresighted decisions by better utilizing system state information and considering the immediate cost as well as the future costs. From Fig 4(b) and Fig 4(c), we can see that, as w increases, the average power costs of the proposed two policies decrease, at the expense of the average delay costs. This reveals the tradeoff between the average delay cost and the average power cost.

In Fig. 5, Fig. 6, and Fig. 7, we investigate the impacts of the Zipf parameter, the cache size of SBSs, and the number of users on the performance of the proposed suboptimal policy and the two baseline policies, respectively. We consider a network with 1 MBS, 2 SBSs and 20 contents. We set $w = 3$, $\mathcal{M}_0 = \mathcal{M}$, $|\mathcal{K}_0| : |\mathcal{K}_1| : |\mathcal{K}_2| = 1 : 2 : 2$, $p(0, m) = 30$ for all $m \in \mathcal{M}_0$, and $p(n, m) = 3$ for $n = 1, 2$ and $m \in \mathcal{M}_n$. Here, $|\mathcal{K}_0| : |\mathcal{K}_1| : |\mathcal{K}_2|$ indicates the ratio of the numbers of the users in \mathcal{K}_0 , \mathcal{K}_1 , and \mathcal{K}_2 . From Fig. 5 - 7, it can be seen that the proposed suboptimal policy outperforms the two baseline policies in terms of the average network cost.

Fig. 5 illustrates the average costs versus the Zipf parameter α for the aforementioned three policies. The α parameter determines the ‘‘skewness’’ of the content popularity distribution, i.e., a large α indicates that a small number of contents account for the majority of content requests. We observe that for the proposed suboptimal policy, as α increases, the average network cost decreases and better delay performance can be achieved with less transmission power. This reveals that the proposed suboptimal policy can utilize caching more effectively as the content popularity distribution gets steeper.

Fig. 6 illustrates the average costs versus the cache size of SBSs $|\mathcal{M}_S|$. For the proposed suboptimal policy, we observe that with the increase of $|\mathcal{M}_S|$, the average network cost decreases (slightly) and the average power cost decreases without sacrificing the delay performance. In addition, for the greedy policy, when $|\mathcal{M}_S|$ increases, the average power cost decreases, however, at the expense of the increase of the average delay and network costs. The reason is that, when more contents are cached in SBSs, the transmission opportunities of SBSs increase and the proposed suboptimal policy can utilize these opportunities more properly than the greedy policy. We also notice that for the greedy policy, when $|\mathcal{M}_S|$ increases, the average network and delay costs increase

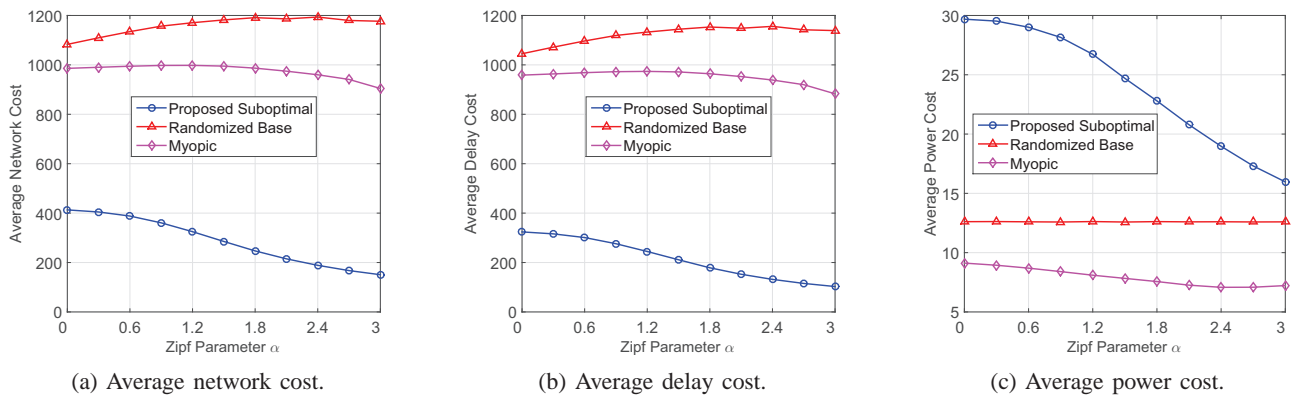


Fig. 5: Average network, delay, and power costs versus Zipf parameter α . $K = 30$ and $|\mathcal{M}_1| = |\mathcal{M}_2| = 10$.

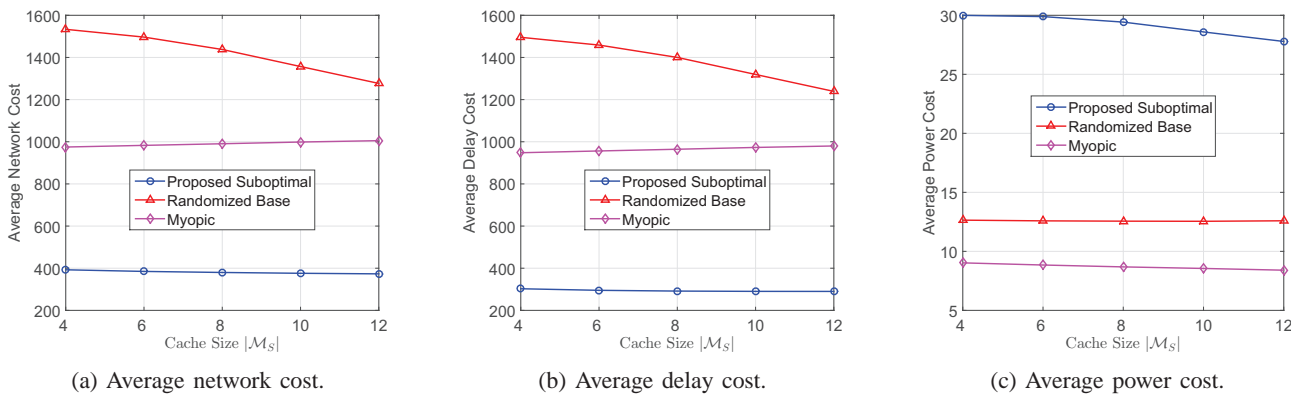


Fig. 6: Average network, delay, and power costs versus cache size of SBSs $|\mathcal{M}_S|$. $|\mathcal{M}_S| = |\mathcal{M}_1| = |\mathcal{M}_2|$, $K = 30$, and $\alpha = 0.75$.

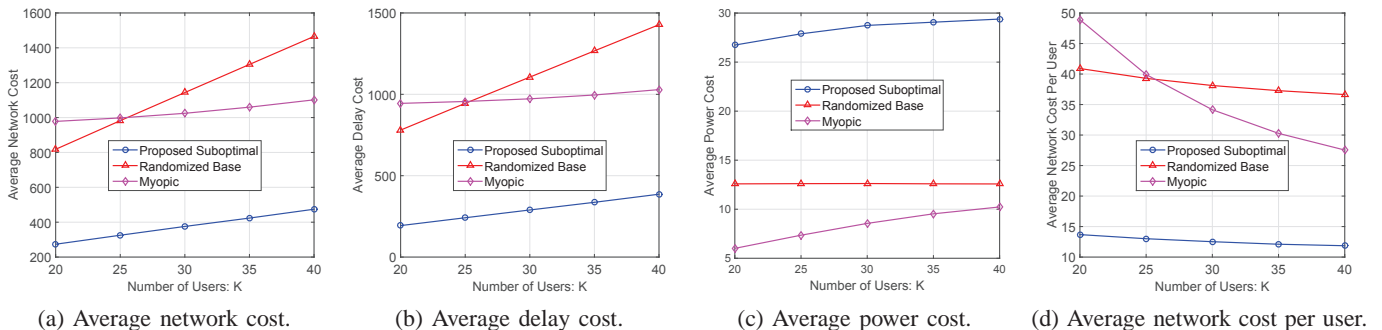


Fig. 7: Average network cost, delay cost, power cost, and network cost per user versus number of users K . $\alpha = 0.75$ and $|\mathcal{M}_1| = |\mathcal{M}_2| = 10$.

and the average power cost decreases. The intuitive reasons are as follows. First, the cost function $C(\mathbf{Q}, \mathbf{u})$ is not equivalent to the original objective function in Problem 1. Second, under the simulation settings, when $|\mathcal{M}_S|$ increases, i.e., the transmission opportunities of the SBSs increase, the greedy policy would be more likely to schedule the SBSs, which leads to certain reduction of the average power cost but much larger increase of the average delay cost, i.e., the increase of the average network cost. Therefore, this reveals that the greedy policy could not properly utilize the transmission opportunities offered by the increase of cache sizes.

Fig. 7 illustrates the average costs versus the number of users K . From Fig. 7(a)-7(c), it can be seen that with the

increase of K , the average network, delay, and power costs of the proposed suboptimal policy increase. The reason is that, when the average request arrival rate increases (as K increases), there will be more requests waiting for service and the BSs are more willing to operate instead of keeping idle. From Fig. 7(d), we can see that with the increase of K , the average network costs per user of all policies decrease. This reflects the benefit of multicast.

Next, we compare the computational complexity of the standard optimal algorithm (PIA), the proposed structure-aware optimal algorithm (SPIA) and the proposed low-complexity suboptimal algorithm (SSA). Table II illustrates the computational time comparison for a network with 1 MBS, 1 SBS

TABLE II: Average computation time (sec) for different algorithms.

Algorithms		$ \mathcal{M}_0 = 2, \mathcal{M}_1 = 1$	$ \mathcal{M}_0 = 3, \mathcal{M}_1 = 1$	$ \mathcal{M}_0 = 3, \mathcal{M}_1 = 2$
PIA: [32, Chapter 8.6]	evaluation	3.283	28.955	330.88
	improvement	2.103	19.136	187.73
	total	5.386	48.091	518.61
SPIA: Algorithm 1	evaluation	3.288	28.436	318.63
	improvement	1.133	8.887	88.360
	total	4.421	37.323	406.99
SSA: Algorithm 2	step 1	0.008	0.015	0.054
	step 2	0.223	3.859	36.240
	total	0.231	3.874	36.294

and 4 users. The simulation is carried out on a Windows x86 machine with a dual-core 2.93 GHz Intel processor and 4GB RAM using Python 3. We set $|\mathcal{K}_0| = |\mathcal{K}_1| = 2$, $w = 1$, $\alpha = 0.75$, $N_{n,m} = 4$ for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. It can be seen that SPIA has much lower computational complexity than PIA, with a reduction of about 50% in the policy improvement step and a reduction of about 20% in the total algorithm. Moreover, we observe that the computation times of PIA and SPIA grow rapidly with the cardinality of the system state space ($|\mathbf{Q}| = \prod_{n \in \mathcal{N}} \prod_{m \in \mathcal{M}_n} |\mathcal{Q}_{n,m}| = 5^{|\mathcal{M}_0| + |\mathcal{M}_1|}$), while the computation time of SSA grows almost linearly with $|\mathbf{Q}|$. The computational savings of SSA compared with PIA and SPIA are significant. These verify the discussions in Section V and Section VI.

VIII. CONCLUSION

In this paper, we study optimal content delivery strategy in a cache-enabled HetNet by taking into account the inherent multicast capability of wireless medium. We establish a content-centric request queue model and then formulate a stochastic content multicast scheduling problem to jointly minimize the average network delay and power costs under a multiple access constraint. This stochastic optimization problem is an infinite horizon average cost MDP. We show that the optimal multicast scheduling policy, which is adaptive to the request queue state, is of threshold type. Then, we propose a structure-aware optimal algorithm to obtain the optimal policy by exploiting its structural properties. To further reduce the complexity, we propose a low-complexity suboptimal policy, which has similar structural properties to the optimal policy, and develop a low-complexity algorithm to obtain this policy.

This work opens up several directions for future research. First, this work focuses on multicast scheduling for given cache placement. It would be interesting to investigate the joint optimal design of content delivery and cache placement/replacement. Second, in this work, IRM is used to model the content request arrivals. It is of particular importance to consider more practical request traffic models. Finally, in this work, we assume that the coverage areas of the SBSs are disjoint and the SBSs can operate concurrently without mutual interference. It is also of interest to take into account the interference management in cache-enabled HetNets under more general topology and interference models.

APPENDIX A: PROOF OF LEMMA 1

By Proposition 4.2.5 in [19], the Weak Accessibly (WA) condition holds for unichain policies. Thus, by Proposition

4.2.3 in [19], the optimal average network cost of the MDP in Problem 1 is the same for all initial states. In addition, by Proposition 4.2.1 in [19], we know that the solution $(\theta, \{V(\mathbf{Q})\})$ to the following Bellman equation exists.

$$\theta + V(\mathbf{Q}) = \min_{\mathbf{u} \in \mathcal{U}} \left\{ g(\mathbf{Q}, \mathbf{u}) + \sum_{\mathbf{Q}' \in \mathcal{Q}} \Pr[\mathbf{Q}' | \mathbf{Q}, \mathbf{u}] V(\mathbf{Q}') \right\}, \quad \forall \mathbf{Q} \in \mathcal{Q}. \quad (30)$$

By substituting the transition probability $\Pr[\mathbf{Q}' | \mathbf{Q}, \mathbf{u}]$ given in (5) into (30), we have (10), which completes the proof. ■

APPENDIX B: PROOF OF LEMMA 2

We prove Lemma 2 using RVIA and mathematical induction.

First, we introduce RVIA [19, Chapter 4.3]. For each state $\mathbf{Q} \in \mathcal{Q}$, let $V_l(\mathbf{Q})$ be the value function in the l -th iteration, where $l = 0, 1, \dots$. Define the state-action cost function in the l -th iteration as

$$J_{l+1}(\mathbf{Q}, \mathbf{u}) \triangleq g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}[V_l(\mathbf{Q}')], \quad (31)$$

where $g(\mathbf{Q}, \mathbf{u})$ and \mathbf{Q}' are given in (8) and Lemma 1, respectively. Note that $J_{l+1}(\mathbf{Q}, \mathbf{u})$ is related to the R.H.S. of the Bellman equation in (10). For each \mathbf{Q} , RVIA calculates $V_{l+1}(\mathbf{Q})$ according to

$$V_{l+1}(\mathbf{Q}) = \min_{\mathbf{u} \in \mathcal{U}} J_{l+1}(\mathbf{Q}, \mathbf{u}) - \min_{\mathbf{u} \in \mathcal{U}} J_{l+1}(\mathbf{Q}^\dagger, \mathbf{u}), \quad \forall l, \quad (32)$$

where $\mathbf{Q}^\dagger \in \mathcal{Q}$ is some fixed state. Under any initialization of $V_0(\mathbf{Q})$, the generated sequence $\{V_l(\mathbf{Q})\}$ converges to $V(\mathbf{Q})$ [19, Proposition 4.3.2], i.e.,

$$\lim_{l \rightarrow \infty} V_l(\mathbf{Q}) = V(\mathbf{Q}), \quad \forall \mathbf{Q} \in \mathcal{Q}, \quad (33)$$

where $V(\mathbf{Q})$ satisfies the Bellman equation in (10). Let $\mu_l^*(\mathbf{Q})$ denote the control that attains the minimum of the first term in (32) in the l -th iteration for all \mathbf{Q} , i.e.,

$$\mu_l^*(\mathbf{Q}) = \arg \min_{\mathbf{u} \in \mathcal{U}} J_{l+1}(\mathbf{Q}, \mathbf{u}), \quad \forall \mathbf{Q} \in \mathcal{Q}. \quad (34)$$

Define $\mu_l^*(\mathbf{Q}) \triangleq (\mu_{l,n}^*(\mathbf{Q}))_{n \in \mathcal{N}}$, where $\mu_{l,n}^*(\mathbf{Q})$ denotes the control action of BS n for state \mathbf{Q} . We refer to μ_n^* as the optimal policy for the l -th iteration.

Next, we prove Lemma 2 through induction using RVIA. Denote $\mathbf{Q}^1 \triangleq (Q_{n,m}^1)_{n \in \mathcal{N}, m \in \mathcal{M}_n}$ and $\mathbf{Q}^2 \triangleq (Q_{n,m}^2)_{n \in \mathcal{N}, m \in \mathcal{M}_n}$. To prove Lemma 2, it is sufficient to show that for any $\mathbf{Q}^1, \mathbf{Q}^2 \in \mathcal{Q}$ such that $\mathbf{Q}^2 \succeq \mathbf{Q}^1$,

$$V_l(\mathbf{Q}^2) \geq V_l(\mathbf{Q}^1), \quad (35)$$

holds for all $l = 0, 1, \dots$. Note that, \mathbf{Q}^1 and \mathbf{Q}^2 can lie on the boundaries of the state space \mathcal{Q} .

First, we initialize $V_0(\mathbf{Q}) = 0$ for all $\mathbf{Q} \in \mathcal{Q}$. Thus, we have $V_0(\mathbf{Q}^1) = V_0(\mathbf{Q}^2) = 0$, i.e., (35) holds for $l = 0$. Assume that (35) holds for some $l > 0$. We will prove that (35) also holds for $l + 1$. By (32), we have

$$\begin{aligned} V_{l+1}(\mathbf{Q}^1) &= J_{l+1}(\mathbf{Q}^1, \mu_l^*(\mathbf{Q}^1)) - J_{l+1}(\mathbf{Q}^\dagger, \mu_l^*(\mathbf{Q}^\dagger)) \\ &\stackrel{(a)}{\leq} J_{l+1}(\mathbf{Q}^1, \mu_l^*(\mathbf{Q}^2)) - J_{l+1}(\mathbf{Q}^\dagger, \mu_l^*(\mathbf{Q}^\dagger)) \\ &\stackrel{(b)}{=} \mathbb{E}[V_l(\mathbf{Q}^1)] + d(\mathbf{Q}^1) + wp(\mu_l^*(\mathbf{Q}^2)) - J_{l+1}(\mathbf{Q}^\dagger, \mu_l^*(\mathbf{Q}^\dagger)), \end{aligned} \quad (36)$$

where (a) follows from the optimality of $\mu_l^*(\mathbf{Q}^1)$ for \mathbf{Q}^1 in the l -th iteration, (b) directly follows from (31), and $\mathbf{Q}^{1'} = (Q_{n,m}^{1'})_{n \in \mathcal{N}, m \in \mathcal{M}_n}$ with $Q_{0,m}^{1'} \triangleq \min\{\mathbf{1}(\mu_{l,0}^*(\mathbf{Q}^2) \neq m)Q_{0,m}^1 + \tilde{A}_{0,m}, N_{0,m}\}$ for all $m \in \mathcal{M}_0$ and $Q_{n,m}^{1'} \triangleq \min\{\mathbf{1}(\mu_{l,0}^*(\mathbf{Q}^2) \neq m \& \mu_{l,n}^*(\mathbf{Q}^2) \neq m)Q_{n,m}^1 + A_{n,m}, N_{n,m}\}$ for all $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. By (31) and (32), we also have

$$\begin{aligned} V_{l+1}(\mathbf{Q}^2) &= J_{l+1}(\mathbf{Q}^2, \mu_l^*(\mathbf{Q}^2)) - J_{l+1}(\mathbf{Q}^\dagger, \mu_l^*(\mathbf{Q}^\dagger)) \\ &= \mathbb{E}[V_l(\mathbf{Q}^2)] + d(\mathbf{Q}^2) + wp(\mu_l^*(\mathbf{Q}^2)) - J_{l+1}(\mathbf{Q}^\dagger, \mu_l^*(\mathbf{Q}^\dagger)), \end{aligned} \quad (37)$$

where $\mathbf{Q}^{2'} = (Q_{n,m}^{2'})_{n \in \mathcal{N}, m \in \mathcal{M}_n}$ with $Q_{0,m}^{2'} \triangleq \min\{\mathbf{1}(\mu_{l,0}^*(\mathbf{Q}^2) \neq m)Q_{0,m}^2 + \tilde{A}_{0,m}, N_{0,m}\}$ for all $m \in \mathcal{M}_0$ and $Q_{n,m}^{2'} \triangleq \min\{\mathbf{1}(\mu_{l,0}^*(\mathbf{Q}^2) \neq m \& \mu_{l,n}^*(\mathbf{Q}^2) \neq m)Q_{n,m}^2 + A_{n,m}, N_{n,m}\}$ for all $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. Then, we compare (36) and (37) term by term.

Due to $\mathbf{Q}^2 \succeq \mathbf{Q}^1$, we have $\mathbf{Q}^{2'} \succeq \mathbf{Q}^{1'}$, implying that $\mathbb{E}[V_l(\mathbf{Q}^{2'})] \geq \mathbb{E}[V_l(\mathbf{Q}^{1'})]$ by the induction hypothesis. In addition, since $d(\mathbf{Q})$ is a monotonically non-decreasing function of \mathbf{Q} , we have $d(\mathbf{Q}^2) \geq d(\mathbf{Q}^1)$. Thus, we have $V_{l+1}(\mathbf{Q}^2) \geq V_{l+1}(\mathbf{Q}^1)$, i.e., (35) holds for $l + 1$. Therefore, by induction, we can show that (35) holds for any l . By taking limits on both sides of (35) and by (33), we complete the proof of Lemma 2. \blacksquare

APPENDIX C: PROOF OF LEMMA 3

First, we derive the general relationship between $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^1)$ and $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^2)$ for any $\mathbf{u}, \mathbf{v} \in \mathcal{U}$ and any $\mathbf{Q}^1, \mathbf{Q}^2 \in \mathcal{Q}$, which will be used to prove the two properties in Lemma 3. By (13), for any $\mathbf{u}, \mathbf{v} \in \mathcal{U}$ and any $\mathbf{Q}^1, \mathbf{Q}^2 \in \mathcal{Q}$, we have

$$\begin{aligned} &\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^1) - \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^2) \\ &= \left(d(\mathbf{Q}^1) + wp(\mathbf{u}) + \mathbb{E}[V(\mathbf{Q}^1, \mathbf{u})] - d(\mathbf{Q}^1) - wp(\mathbf{v}) \right. \\ &\quad \left. - \mathbb{E}[V(\mathbf{Q}^1, \mathbf{v})] \right) - \left(d(\mathbf{Q}^2) + wp(\mathbf{u}) + \mathbb{E}[V(\mathbf{Q}^2, \mathbf{u})] \right. \\ &\quad \left. - d(\mathbf{Q}^2) - wp(\mathbf{v}) - \mathbb{E}[V(\mathbf{Q}^2, \mathbf{v})] \right) \\ &= \mathbb{E}[V(\mathbf{Q}^1, \mathbf{u})] - \mathbb{E}[V(\mathbf{Q}^1, \mathbf{v})] - \mathbb{E}[V(\mathbf{Q}^2, \mathbf{u})] + \mathbb{E}[V(\mathbf{Q}^2, \mathbf{v})], \end{aligned} \quad (38)$$

where

$$\begin{cases} Q_{0,m}^{1,\mathbf{u}} = \min\{\mathbf{1}(u_0 \neq m)Q_{0,m}^1 + \tilde{A}_{0,m}, N_{0,m}\}, \forall m \in \mathcal{M}_0 \\ Q_{n,m}^{1,\mathbf{u}} = \min\{\mathbf{1}(u_0 \neq m \& u_n \neq m)Q_{n,m}^1 + A_{n,m}, N_{n,m}\}, \\ \quad \forall n \in \mathcal{N}^+, \forall m \in \mathcal{M}_n \end{cases} \quad (39a)$$

$$\begin{cases} Q_{0,m}^{1,\mathbf{v}} = \min\{\mathbf{1}(v_0 \neq m)Q_{0,m}^1 + \tilde{A}_{0,m}, N_{0,m}\}, \forall m \in \mathcal{M}_0 \\ Q_{n,m}^{1,\mathbf{v}} = \min\{\mathbf{1}(v_0 \neq m \& v_n \neq m)Q_{n,m}^1 + A_{n,m}, N_{n,m}\}, \\ \quad \forall n \in \mathcal{N}^+, \forall m \in \mathcal{M}_n \end{cases} \quad (39b)$$

$$\begin{cases} Q_{0,m}^{2,\mathbf{u}} = \min\{\mathbf{1}(u_0 \neq m)Q_{0,m}^2 + \tilde{A}_{0,m}, N_{0,m}\}, \forall m \in \mathcal{M}_0 \\ Q_{n,m}^{2,\mathbf{u}} = \min\{\mathbf{1}(u_0 \neq m \& u_n \neq m)Q_{n,m}^2 + A_{n,m}, N_{n,m}\}, \\ \quad \forall n \in \mathcal{N}^+, \forall m \in \mathcal{M}_n \end{cases} \quad (39c)$$

$$\begin{cases} Q_{0,m}^{2,\mathbf{v}} = \min\{\mathbf{1}(v_0 \neq m)Q_{0,m}^2 + \tilde{A}_{0,m}, N_{0,m}\}, \forall m \in \mathcal{M}_0 \\ Q_{n,m}^{2,\mathbf{v}} = \min\{\mathbf{1}(v_0 \neq m \& v_n \neq m)Q_{n,m}^2 + A_{n,m}, N_{n,m}\}, \\ \quad \forall n \in \mathcal{N}^+, \forall m \in \mathcal{M}_n \end{cases} \quad (39d)$$

Next, based on (38), we prove Property 1 in Lemma 3. Suppose action \mathbf{u} satisfies that $u_0 = j \in \mathcal{M}_0$, and \mathbf{Q}^1 and \mathbf{Q}^2 satisfy the following relation:

$$\begin{cases} Q_{n,m}^1 \geq Q_{n,m}^2 & \text{if } n \in \{0\} \cup \mathcal{N}_j \text{ and } m = j, \\ Q_{n,m}^1 = Q_{n,m}^2 & \text{otherwise.} \end{cases} \quad (40)$$

for each $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. By comparing (39a) with (39c), we can see that $Q_{n,m}^{1,\mathbf{u}} = Q_{n,m}^{2,\mathbf{u}}$ for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$, i.e., $\mathbf{Q}^{1,\mathbf{u}} = \mathbf{Q}^{2,\mathbf{u}}$. Thus, we have $\mathbb{E}[V(\mathbf{Q}^{1,\mathbf{u}})] = \mathbb{E}[V(\mathbf{Q}^{2,\mathbf{u}})]$. By comparing (39b) with (39d), we can see that $\mathbf{Q}^{1,\mathbf{v}} \succeq \mathbf{Q}^{2,\mathbf{v}}$. Thus, by Lemma 2, we have $\mathbb{E}[V(\mathbf{Q}^{1,\mathbf{v}})] \geq \mathbb{E}[V(\mathbf{Q}^{2,\mathbf{v}})]$. Therefore, by (38), we have $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^1) \leq \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^2)$, which completes the proof of Property 1 in Lemma 3.

Finally, based on (38), we prove Property 2 in Lemma 3. Suppose action \mathbf{u} satisfies that $u_i = j \in \mathcal{M}_i$ for some $j \in \mathcal{N}^+$, and \mathbf{Q}^1 and \mathbf{Q}^2 satisfy the following relation:

$$\begin{cases} Q_{n,m}^1 \geq Q_{n,m}^2 & \text{if } n = i \text{ and } m = j, \\ Q_{n,m}^1 = Q_{n,m}^2 & \text{otherwise.} \end{cases} \quad (41)$$

for each $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. Similarly, by comparing (39a) with (39c) and comparing (39b) with (39d), we can see that $\mathbf{Q}^{1,\mathbf{u}} = \mathbf{Q}^{2,\mathbf{u}}$ and $\mathbf{Q}^{1,\mathbf{v}} \succeq \mathbf{Q}^{2,\mathbf{v}}$. Therefore, by Lemma 2 and (38), we have $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^1) \leq \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^2)$, which completes the proof of Property 2 in Lemma 3. \blacksquare

APPENDIX D: PROOF OF THEOREM 1

We first prove Property 1 of Theorem 1. Consider multicast scheduling action $\mathbf{u} = \mathbf{0}$, BS $n \in \mathcal{N}$, content $m \in \mathcal{M}_n$, another action $\mathbf{v} = (v_i)_{i \in \mathcal{N}} \in \mathcal{U}$ where $v_n = m$, and request queue state $\mathbf{Q} = (Q_{i,j})_{i \in \mathcal{N}, j \in \mathcal{M}_i} \in \mathcal{Q}$ where $Q_{n,m} = \phi_0^+(\mathbf{Q}_{-n,-m})$. Note that, if $\phi_0^+(\mathbf{Q}_{-n,-m}) = -\infty$, Property 1 of Theorem 1 always holds. Therefore, in the following, we only consider $\phi_0^+(\mathbf{Q}_{-n,-m}) > -\infty$. According to the definition of $\phi_{\mathbf{u}}^+(\mathbf{Q}_{-n,-m})$ in (15), we can see that, $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0$, i.e., \mathbf{u} dominates \mathbf{v} at state \mathbf{Q} . Now consider another queue state $\mathbf{Q}' = (Q'_{i,j})_{i \in \mathcal{N}, j \in \mathcal{M}_i} \in \mathcal{Q}$ where $Q'_{n,m} \leq Q_{n,m}$ and $Q'_{i,j} = Q_{i,j}$ for all $(i,j) \neq (n,m)$. Since

$\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) = -\Delta_{\mathbf{v},\mathbf{u}}(\mathbf{Q})$, by Lemma 3, we know that $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ is monotonically non-decreasing with $Q_{n,m}$. Thus, we have

$$\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}') \leq \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0, \quad (42)$$

i.e., \mathbf{u} dominates \mathbf{v} at state \mathbf{Q}' . By the definition of \mathcal{Q}_0 , we can see that if $\mathbf{Q} \in \mathcal{Q}_0$, \mathbf{u} dominates all $\mathbf{v} \in \mathcal{U}$ and $\mathbf{v} \neq \mathbf{u}$ at state \mathbf{Q} , i.e., $\mu^*(\mathbf{Q}) = \mathbf{0}$. We complete the proof of Property 1 in Theorem 1.

Next, we prove Property 2 of Theorem 1. Consider BS $n \in \mathcal{N}$, content $m \in \mathcal{M}_n$, multicast scheduling action $\mathbf{u} = (u_i)_{i \in \mathcal{N}} \in \mathcal{U}$ where $u_n = m$, and request queue state $\mathbf{Q} = (Q_{i,j})_{i \in \mathcal{N}, j \in \mathcal{M}_i} \in \mathcal{Q}$ where $Q_{n,m} = \phi_{\mathbf{u}}^-(\mathbf{Q}_{-n,-m})$. Similar to the proof of Property 1 of Theorem 1, we only need to consider that $\phi_{\mathbf{u}}^-(\mathbf{Q}_{-n,-m}) < +\infty$. According to the definition of $\phi_{\mathbf{u}}^-(\mathbf{Q}_{-n,-m})$ in (15), we can see that, $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0$ for all $\mathbf{v} \in \mathcal{U}$ and $\mathbf{v} \neq \mathbf{u}$, i.e., $\mu^*(\mathbf{Q}) = \mathbf{u}$.

- We first prove the first part of Property 2. Consider queue state $\mathbf{Q}^1 = (Q_{i,j}^1)_{i \in \mathcal{N}, j \in \mathcal{M}_i} \in \mathcal{Q}$ where $Q_{n,m}^1 \geq Q_{n,m}$ and $Q_{i,j}^1 = Q_{i,j}$ for all $(i,j) \neq (n,m)$. To prove the first part of Property 2, it is equivalent to show that $\mu^*(\mathbf{Q}^1) = \mathbf{u}$. By Lemma 3, for all $\mathbf{v} \in \mathcal{U}$ and $\mathbf{v} \neq \mathbf{u}$, we have

$$\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^1) \leq \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0, \quad (43)$$

i.e., $\mu^*(\mathbf{Q}^1) = \mathbf{u}$. We complete the proof of the first part in Property 2.

- Then, we prove the second part of Property 2, i.e., the monotonically non-increasing property of $\phi_{\mathbf{u}}^-(\mathbf{Q}_{-0,-m})$ in terms of $Q_{n,m}$ for all $n \in \mathcal{N}_m$. Consider another queue state $\mathbf{Q}^2 = (Q_{i,j}^2)_{i \in \mathcal{N}, j \in \mathcal{M}_i} \in \mathcal{Q}$ where $Q_{i,j}^2 \geq Q_{i,j}$ if $i \in \mathcal{N}_j$ and $j = m$, and $Q_{i,j}^2 = Q_{i,j}$ otherwise. To prove the second part of Property 2, it is equivalent to show that $\mu^*(\mathbf{Q}^2) = \mathbf{u}$. By Lemma 3, for all $\mathbf{v} \in \mathcal{U}$ and $\mathbf{v} \neq \mathbf{u}$, we have

$$\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}^2) \leq \Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0, \quad (44)$$

i.e., $\mu^*(\mathbf{Q}^2) = \mathbf{u}$. We complete the proof of Property 2 in Theorem 1. ■

APPENDIX E: PROOF OF LEMMA 4

Along the line of the proof of Lemma 3 in [33], we prove the additive property of the value function. Note that, we have $g(\mathbf{Q}, \mathbf{u}) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} g_{n,m}(Q_{n,m}, \mathbf{u})$. In addition, by the relation between the joint distribution and the marginal distribution, we have $\sum_{\mathbf{Q} \in \mathcal{Q}} \Pr[\mathbf{Q}' | \mathbf{Q}, \mathbf{u}] = \sum_{Q'_{n,m} \in \mathcal{Q}_{n,m}} \Pr[Q'_{n,m} | \mathbf{Q}, \mathbf{u}] = \sum_{Q'_{n,m} \in \mathcal{Q}_{n,m}} \Pr[Q'_{n,m} | Q_{n,m}, \mathbf{u}]$. Therefore, by substituting $\hat{\theta} = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \hat{\theta}_{n,m}$ and $\hat{V}(\mathbf{Q}) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_n} \hat{V}_{n,m}(Q_{n,m})$ into (22), we can see that the equality in (23) holds, which completes the proof. ■

APPENDIX F: PROOF OF THEOREM 2

First, we show that for all $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$, the per-BS-content value function $\hat{V}_{n,m}(Q_{n,m})$ satisfies

$$\hat{V}_{n,m}(Q_{n,m}^2) \geq \hat{V}_{n,m}(Q_{n,m}^1), \quad (45)$$

for any $Q_{n,m}^1, Q_{n,m}^2 \in \mathcal{Q}_{n,m}$ such that $Q_{n,m}^2 \geq Q_{n,m}^1$. Similar to the proof of Lemma 2, we define:

$$\begin{aligned} \hat{J}_{n,m}^{l+1}(Q_{n,m}) &\triangleq \mathbb{E}^{\hat{\mu}} [g_{n,m}(Q_{n,m}, \mathbf{u})] \\ &+ \sum_{Q'_{n,m} \in \mathcal{Q}_{n,m}} \mathbb{E}^{\hat{\mu}} [\Pr[Q'_{n,m} | Q_{n,m}, \mathbf{u}]] \hat{V}_{n,m}^l(Q'_{n,m}), \end{aligned} \quad (46)$$

where $\hat{V}_{n,m}^l(Q_{n,m})$ denotes the per-BS-content value function in the l -th iteration. For each $Q_{n,m} \in \mathcal{Q}_{n,m}$, RVIA calculates $\hat{V}_{n,m}^{l+1}(Q_{n,m})$ according to:

$$\hat{V}_{n,m}^{l+1}(Q_{n,m}) = \hat{J}_{n,m}^{l+1}(Q_{n,m}) - \hat{J}_{n,m}^{l+1}(Q_{n,m}^\dagger), \quad (47)$$

where $Q_{n,m}^\dagger \in \mathcal{Q}_{n,m}$ is some fixed state. Following the proof of Lemma 2, we can prove that for any $Q_{n,m}^1, Q_{n,m}^2 \in \mathcal{Q}_{n,m}$ such that $Q_{n,m}^2 \geq Q_{n,m}^1$, $\hat{V}_{n,m}^l(Q_{n,m}^2) \geq \hat{V}_{n,m}^l(Q_{n,m}^1)$ holds for all $l = 0, 1, \dots$. Thus, we can show that (45) holds through induction using RVIA. Then, by (45), we can show that $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ possesses the same monotonicity properties of $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$, following the proof of Lemma 3. Finally, by using the monotonicity properties of $\hat{\Delta}_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$, we can show the structural properties of $\hat{\mu}^*$, following the proof of Theorem 1. We complete the proof. ■

REFERENCES

- [1] B. Zhou, Y. Cui, and M. Tao, "Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks," in *ACM CoNEXT 2015 Workshop on CCDWN*, Dec. 2015.
- [2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019," *White Paper*, Feb 2015.
- [3] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Vitsosky, T. Thomas, J. Andrews, P. Xia, H. Jo, H. Dhillon, and T. Novlan, "Heterogeneous cellular networks: From theory to practice," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 54–64, June 2012.
- [4] A. F. Molisch, G. Caire, D. Ott, J. R. Foerster, D. Bethanabhotla, and M. Ji, "Caching eliminates the wireless bottleneck in video aware wireless networks," *Advances in Electrical Engineering*, vol. 2014, 2014.
- [5] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug 2014.
- [6] G. Paschos, E. Baştug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *arXiv preprint arXiv:1602.00173*, 2016.
- [7] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," in *Proc. IEEE ISIT*, June 2015, pp. 809–813.
- [8] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, Dec 2013.
- [9] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. IEEE INFOCOM*, April 2015.
- [10] A. Liu and V. Lau, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan 2014.
- [11] E. Bastug, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: modeling and tradeoffs," *EURASIP Journal of Wireless Communications and Networking*, vol. 2015, p. 1, 2015.
- [12] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, pp. 1–1, 2015.
- [13] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, 2012.
- [14] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [15] U. Niesen and M. Maddah-Ali, "Coded caching for delay-sensitive content," in *Proc. IEEE ICC*, June 2015.

- [16] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995–3007, April 2016.
- [17] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless networks with elastic and inelastic traffic," *IEEE/ACM Trans. New.*, vol. 22, no. 3, pp. 864–874, June 2014.
- [18] B. Zhou, Y. Cui, and M. Tao, "Optimal dynamic multicast scheduling for cache-enabled content-centric wireless networks," in *Proc. IEEE ISIT*, June 2015.
- [19] D. P. Bertsekas, *Dynamic programming and optimal control, 3rd edition, volume II*. Belmont, MA: Athena Scientific, 2011.
- [20] M. Shifrin, R. Atar, and I. Cidon, "Optimal scheduling in the hybrid-cloud," in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 51–59.
- [21] M. Shifrin, A. Cohen, O. Gurewitz, and O. Weisman, "Coded retransmission in wireless networks via abstract MDPs: Theory and algorithms," *arXiv preprint arXiv:1502.02893*, 2015.
- [22] E. Altman, R. El-Azouzi, D. S. Menasche, and Y. Xu, "Forever young: Aging control for smartphones in hybrid networks," *arXiv preprint arXiv:1009.4733*, 2010.
- [23] C. H. Xia, G. Michailidis, N. Bambos, and P. W. Glynn, "Optimal control of parallel queues with batch service," *Probability in the Engineering and Informational Sciences*, vol. 16, no. 03, pp. 289–307, 2002.
- [24] R. Gummadi, "Optimal control of a broadcasting server," in *Proc. 48th IEEE Conf. Decision Control (CDC/CCC)*, Dec. 2009.
- [25] D. V. Djonin and V. Krishnamurthy, "MIMO transmission control in fading channels: a constrained markov decision process formulation with monotone randomized policies," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, 2007.
- [26] A. H. Elwany, N. Z. Gebrael, and L. M. Maillart, "Structured replacement policies for components with complex degradation processes and dedicated sensors," *Oper. Res.*, vol. 59, no. 3, pp. 684–695, 2011.
- [27] E. J. Rosensweig, D. S. Menasche, and J. Kurose, "On the steady-state of cache networks," in *Proc. IEEE INFOCOM*, April 2013.
- [28] K. Deb, "Multi-objective optimization," in *Search methodologies*. Springer, 2014, pp. 403–449.
- [29] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.
- [30] M. Levorato, U. Mitra, and M. Zorzi, "On optimal control of wireless networks with multiuser detection, hybrid ARQ and distortion constraints," in *Proc. IEEE INFOCOM*, April 2009.
- [31] G. Koole, "Monotonicity in markov reward and decision chains: Theory and applications," *Foundations and Trends in Stochastic Systems*, vol. 1, no. 1, pp. 1–76, 2006.
- [32] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. New York, NY, USA: Wiley, 2009, vol. 414.
- [33] Y. Cui, V. Lau, and Y. Wu, "Delay-aware BS discontinuous transmission control and user scheduling for energy harvesting downlink coordinated MIMO systems," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3786–3795, July 2012.
- [34] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM*, March 1999.
- [35] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.