Joint Optimal Multicast Scheduling and Caching for Improved Performance and Energy Saving in Wireless Heterogeneous Networks

Lujie Zhong, Changqiao Xu, *Senior Member, IEEE,* Jiewei Chen, Weiqi Yan, Shujie Yang, *Member, IEEE,* Gabriel-Miro Muntean, *Senior Member, IEEE*

***Abstract*—Base station caching and multicast are two promising methods to support mass content delivery in future wireless network environments. However, existing scheduling designs do not take full advantage of the two methods. This paper focuses on employing multicast scheduling and caching in a network architecture which involves both macro cell base stations (MBS) and small cell base stations (SBS) in order to achieve joint optimization of average delay and power consumption. We describe this co-optimization problem as the Multicast-Aware Caching Scheduling Problem (MACSP). This paper proposes a novel pending request queue model, which aims to solve the problem of long waiting time for non-popular content, and transform this collaborative multicast-cache scheduling problem into a Markov Decision Process that can be solved using reinforcement learning methods. For actual deployment, the paper further introduces a Distributed Policy Gradient algorithm (DPG) with similar performance and lower complexity. The simulation-based testing results demonstrate that our model and algorithm have better performance and lower energy consumption than existing state-of-the-art approaches.**

***Index Terms*—Cache; Multicast; Markov Decision Process; Reinforcement Learning.**

I. Introduction

A. Motivation

According to the Cisco Visual Network Index (Cisco VNI), mobile data traffic worldwide is expected to continue to grow at a rate of 45% per year over the next few years due to the large-scale use of smartphones, new broadband services and applications. By 2020, traffic from wireless and mobile devices will account for more than two-thirds of total IP traffic [1]. Not only the amount of data is an issue, but also the requirements for the delivery of data associated with the emerging services, many rich media-based. The fifth-generation mobile network system (5G) has set ambitious performance targets for the immediate future to support these services. However, the 5G network features such as high-speed, multi-access, low latency, etc. have their limitations. There limitations are mostly in terms of energy consumption and efficiency of resource utilization. Different candidate solutions were proposed to address these limitations and two important avenues have involved multicast and caching [2]–[6].

Multicast transmissions can simultaneously support services for a large number of users, which is to some extent to meet the increasing demand for mobile video data and to provide better quality of experience (QoE) for end users [7], [8]. Many operators use multicast to utilize more efficiently the available bandwidth of their network and deliver the same content to multiple users (receivers). For example, multicast is often used to deliver advertising content, specifically to set up mobile ads, download news, stock market reports, and weather at specific locations. At the same time, multicast has been incorporated into the 3GPP specifications for LTE as evolved multimedia broadcast and multicast service (eMBMS) [9]. By using eMBMS, it is possible to fully support broadcast and multicast transmissions in LTE and LTE-A systems. Ericsson and Qualcomm's LTE broadcast solutions are typical commercial examples of eMBMS [10] [11]. This technique uses a common carrier frequency to synchronize the transmission between the sender and the receiver and can be applied to multiple cells. Therefore, multicast only consumes radio resources as required by a unicast service, and the remaining resources can be used to support other transmissions, thereby enhancing the network capacity. The use of systems optimized for unicast services for multicast transmissions can result in performance degradations in terms of spectrum, energy efficiency, and QoE [12]. Although the 3GPP working group reached a consensus to remove the TDM-related constraints in the LTE Release 14 specification to support a more efficient and independent eMBMS network [13] [14], there are still some challenges associated with the scheduling and resource allocation (SRA) process.

Related to caching, the academic community has performed a lot of research on network cache architecture and algorithms. However, with the rapid growth of mobile video services, current network architectures will become increasingly difficult to adapt to the ever-increasing content request rate. Taking full advantage of the fact that end users request popular content and placing popular content closer to the end user can directly reduce the service delay and traffic load of the core network, and indirectly address network congestion problems

This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61872253 and 61871048, by the Science Foundation Ireland (SFI) Research Centres Programme under Grant Number 12/RC/2289 (Insight Centre for Data Analytics) and 16/SP/3804 (ENABLE), by the 111 Project (B18008).

L. Zhong is with Information Engineering College, Capital Normal University, 100048, Beijing, China. E-mail: zhonglj@cnu.edu.cn.

C. Xu, J. Chen, W. Yan and S. Yang are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 100876, Beijing, China. (e-mail: {cqxu,chenjiewei,zzywq,sjyang}@bupt.edu.cn).

G.-M. Muntean is with Performance Eng. Lab, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland. E-mail: gabriel.muntean@dcu.ie.

[15]. Many existing works have utilized this design principle to develop dynamic content caching algorithms and use a network caching architecture. Regardless of the approach, the solutions allocate storage resources near end users, rather than just storing data in the data center. Examples of commercial systems that support caching include Altobridge's "edge data" solution [16], Nokia Siemens Networks' liquid application [17] and Saguna Networks' Open RAN platform [18].

In general, multicast and caching systems are independently designed according to different requirements. However, considering the various mutual benefits a joint design of these two technologies is expected to achieve superior performance in mass content delivery in wireless networks. There have been joint studies on multicast and caching mechanisms in content distribution [19], [20]. Most researchers use multicast or caching as an indicator of optimization, and then limit the conditions of the other technology employment in different scenarios. However, most of the literature does not consider the fact that macro base stations (MBS) coexist with small cell base stations (SBS) and does not exploit the synergy between MBS multicast and SBS caching plus multicast over a long period of time. No solution solves this problem known to be NP-hard, despite some of them making excellent contributions [2], [21], [22].

In theory, when enough content is reused, caching is beneficial. When many users generate requests for a particular content file at the same time, a multicast approach is useful. This is common when crowds gather and are interested in the same content, such as in sports competitions, concerts, and public demonstrations, often with thousands of participants.

It is also known that caching on a small base station (SBS) can reduce backhaul data traffic, shorten transmission delays, and improve the quality of the user experience [23]. However, these studies only consider the caching strategy under unicast transmission of wireless networks, or assume that each SBS will multicast one content per unit time slot only.

B. Contributions

In this paper, we consider a more efficient deployment, in which one piece of content is multicast in MBS, while at the same time, SBS can multicast multiple pieces of content, so that the multicast characteristics of the wireless network can be fully utilized. This approach is expected to be beneficial to reducing network delay and improving user experience, as well as reducing energy consumption.

In order to solve the above problem, we study a cache-enabled 5G network that includes one MBS, N SBSs, and M pieces of content and uses multicast. For a given situation, we consider queuing pending requests and the following problems. The caching problem determines what content each SBS multicasts in a time slot. The multicast problem determines whether all BSs are multicasting in a time slot or not, and if a BS decides to multicast, then what content should be multicast? The main contributions of the paper are as follows:

Time-Cumulative-Markup Method: In order to record pending requests from each user on the network, we model the pending request queues associated with content m and a base station (BS). However, in a 5G scenario, the network size is large, and the content request frequency is high, so that more popular content requests are rapidly accumulated, resulting in a long queue of pending requests. There will also be some less popular content that cannot be accumulated for a period of time, causing the content to be selected for multicasting and wait a long time while competing for pending request queues. In order to address this problem we propose the Time-Cumulative-Markup Method which reduces the wait time for the unpopular content by redesigning the pending request queuing. Simulation-based results demonstrate the effectiveness of this method.

Multicast-Aware Caching Scheduling Problem: The premise of most research on multicasting and caching is in a certain time slot, that is to say, few studies have considered changes in multicasting and caching over a continuous period of time. The authors of [2] proved that the multicast delivery problem is an NP-hard problem. We go a step further and consider this optimization problem across multiple time slots and denote it as the Multicast-Aware Caching Scheduling Problem (MACSP).

The problem targets joint optimization of multicast with caching across multiple time slots in 5G systems, aiming to achieve minimum pending queues and taking energy constraints into account. To the best of our knowledge, we are the first researchers which focus on this optimization across more than one time slot. We employ Markov Decision Process and at the same time, reinforcement learning to find the optimal solution for MACSP. Experimental results show that the proposed algorithm is superior to the greedy algorithm.

Distributed Policy Gradient Algorithm: We propose the Distributed Policy Gradient algorithm (DPG), which combines the use of the Deep Deterministic Policy Gradient (DDPG) and Deep Q-Network (DQN) training methods. DPG is employed into the entire Markov decision-making process. Since the MACSP problem is a complex collaborative problem, solving this problem is also a NP-hard problem. Therefore, we have designed a distributed algorithm that divides SBS and MBS actions into two state spaces for training. The final simulation results show the effectiveness of this algorithm design method, and demonstrates that its convergence is good.

The rest of this article is organized as follows. Section II reviews related work. Section III introduces the system model and problem formulation. Section IV gives the designed reinforcement learning algorithm for solving the optimal solution in the average time of MACSP problem. Section V evaluates the performance of the proposed algorithm through experimental simulation.

II. Related Works

This section discusses multicast and caching-based mechanisms proposed in different scenarios, identifying the scientific and technological gaps related to our research.

A. Research on Caching

Today, content sharing network services are becoming more popular, including on-line social networking (OSN), photo

sharing, and video on demand, which have to manage a large amount of content. An efficient solution for this content management is to deploy a collaborative cache. Specifically, in [24], the authors studied cache placement on a collaborative cache built from a single client cache in an on-line social network or web service, and proposed a client to maintain content and cache content, so that the mapping between the client and the workload statistics can be used to design a cache placement scheme. In [25], the authors consider a K-user cache-assisted wireless multi-antenna symmetric broadcast channel with random fading and imperfect feedback. In this case, the article gets an approximate best solution through which identifies new synergies between the use of code buffer and delayed CSIT. At the same time, intra-network caching is one of the key technologies in the content-centric mobile ad hoc network (CCMAN), which can significantly reduce network traffic load and improve content retrieval performance. In [26], the author analyzes the theoretical performance of the cache in CCMAN, defines and deduces the cache utility, and finally proposes a CSEC scheme to improve the efficiency of cache space utilization in CCMAN. To better improve the performance of content caching, the emerging layered network architecture makes it possible to leverage cloud-centric and edge-centric caching. In [15], the authors propose a cache design of mixed content, which is designed to support the average higher request content data rate latency as much as possible on a limited service basis, they also solve the NP-hard cache control problem by using the Lyapunov optimization method and tight coupling between CU cache and BS cache control decisions. As the same for cloud storage applications, the literature [27] proposes a service curve based QoS algorithm to support latency in the same storage system to ensure application execution, which not only provides QoS guarantees for applications, but also pursues better system utilization. In order to further improve the user's experience of network use, [28] developed the best economic caching solution in the cache-enabled heterogeneous network, while providing mobile users with multimedia video services with personalized viewing quality. Meanwhile the author designs a heuristic algorithm based on greedy strategy to achieve near-optimal layer cache index, and proves the performance superiority of the proposed SVC-based caching scheme.

B. Research on Multicast

In recent years, multicasting data to mobile users (e.g. video streaming, video conferencing, IPTV, distribution of news and alerts, or the purpose of application and operating system updates) has become increasingly important. Since such traffic in cellular networks grows very rapidly and wireless resources are scarce and expensive, improving the efficiency of wireless multicasting is highly practical. In [29], the authors propose a heuristic algorithm for opportunistic multicast in wireless networks, which can best solve the problem of balancing the overall throughput and equalizing the throughput of a single receiver, and is suitable for practical on-line scheduling. The software-defined Network Multicast (SDM) mentioned in [30], on the basis of which the author redesigned DYN-SDM, supplemented SDM at key points, which describes the detailed design on internal traffic and service management process of the ISP. At the same time, the article also introduces a new set of SDN-based network layer mechanism to achieve traffic load balancing and group and network dynamic processing. The ISP network, also used in SDN, literature [31] proposed an extensible multicast group management mechanism based on the network function virtualization method to implement and deploy multicast services on the network edge. In addition, this paper also contributes to a lazy load-balanced multicast (L2BM) routing algorithm for sharing core network capacity in a friendly manner between guaranteed bandwidth multicast traffic and best effort traffic which doesn't require real-time link monitoring, reducing economic costs as well. In terms of improving the spectral efficiency of the system, the literature [12] proposed a novel sub-band CQI-based multicast strategy, which relies on the selection of a more spectrally efficient transmission mode to increase the data rate while also meeting the quality index of specified services. In [32], the authors studied NFV-enabled multicasting in Software Defined Networks (SDN) to maximize network throughput while minimizing the cost of allowed NFV-enabled multicast requests. In order to solve the problem of multicast routing delay in the smart grid, [33] proposed a systematic method, namely betweenness centrality to bandwidth ratio tree (BCBT) approach, which uses the shortest path tree (SPT) multicast routing under light traffic, but when the multicast link becomes congested, it will switch to BCBT multicast routing, thereby alleviating SPT congestion.

All of the above research has promoted the development of multicast and cache in various fields, and greatly improved the overall resource utilization of the network while saving economic costs. The joint design of multicast and cache has become one of the solutions to the problem. Next, under the constraint of MBS and SBS joint multicast, the SBS cache mechanism is determined to seek the shortest pending request queue in one time slot.

III. System Model and Problem Formulation

A. Network Model

As shown in Fig. 1, we study a cache-enabled 5G network that contains one macro cell base station (MBS), N small cell base stations (SBSs), and M contents. It is assumed that the coverage areas of SBS do not intersect. The MBS can be associated with any user in the macro cell network, while the SBS can only be associated with users in its coverage area. The set of request content m is denoted by eq. (1).

$$\mathcal{M} \triangleq \{1, 2, ..., M\}. \tag{1}$$

Each SBS n is equipped with a cache of size S_n bytes $(S_n \geq 0)$, which can be filled by content files retrieved from the core network over the backhaul link. In general, SBS has low caching capabilities and cannot provide services directly to a large number of local users. Conversely, the MBS has enough power to download the content requested by the user. Therefore, the MBS directly multicasts to the user, or the local SBS requests the MBS to cache the content and then multicasts

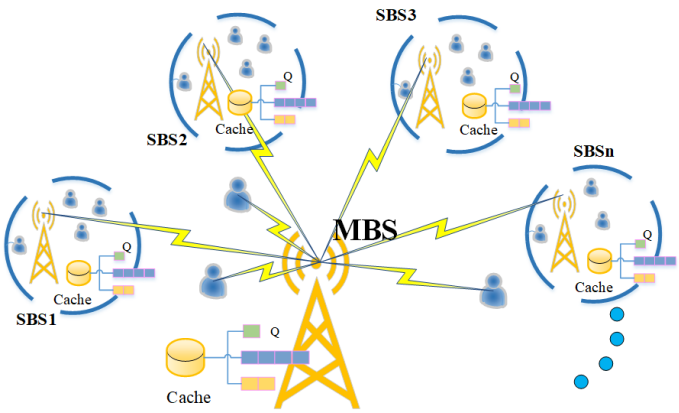


Fig. 1: System Model

to the user. The two modes of the collaborative policy can be used to provide services for the users.

Define a set of all base stations (BSs), denoted as follows:

$$\mathcal{N} \triangleq \{n_0, n_1, n_2, ..., N\}. \tag{2}$$

Especially, n_0 represents MBS, $n \in \mathcal{N}^+ = n_1, n_2, \ldots, N$ express SBS n.

In practice, SBS coverage areas can usually overlap, but each user can be associated with only one SBS according to the best server criteria (e.g. the highest SNR rule). Therefore, it can be assumed that the coverage of the SBS does not coincide.

The operator uses multicast to transmit the same content to multiple receivers. In this case, the users' requests will be aggregated in a short time window and served by a single multicast stream when the corresponding window expires. We use t (unit time) to indicate the duration of the window, also known as the multicast period. The set of time slot allocation without loss of generality is indicated as in eq. (3).

$$\mathcal{T} = \{1, 2, \ldots, T\}. \tag{3}$$

Let time slot t be a multicast cycle of MBS. In each time slot t, MBS only multicasts one content, and SBS n can buffer and multicast multiple contents. The average demand for content m by the users associated with SBS n is represented by $k \geq 0$ (the number of requests at time slot t). The probability that the users of its local SBS coverage area request the content m during the multicast period is denoted by $p_{n,m}$. Similarly, $p_{0,m}$ represents the probability that the user does not request the content m in the coverage area of any SBSs. For example, if the number of requests for content m associated with SBS n follows a *ZipF distribution* with a rate parameter of k, the probability of users' requests is expressed as in eq. (4).

$$p_{n,m}(k, a) = \frac{1}{\zeta(a)k^a}, \tag{4}$$

where a is fixed parameter. Compared to SBS, MBS typically consumes more power, and its power consumption depends on the size of the requested content, channel conditions, and distance between the transmitter and receiver. Let P_n (unit: Watt) represent the minimum transmit power required by the MBS to send a file to the user. According to SINR [2], P_n can be expressed as in the formula from eq. (5).

$$P_n = \delta_s - G_n - G_{n_0} + L_{n_0,i} + \Psi_N + 10\log_{10} B_N \tag{5}$$

In eq. (5), δ_s is the receiver sensitivity of the specific service object, parameter G_n represents the antenna gain of the user in the SBS n coverage area, and G_{n_0} represents the antenna gain of the MBS. $L_{n_0,i}$ is the path loss between MBS and user i, which depends on the channel characteristics and the distance between the MBS and the user, Ψ_N is the shadow component derived from the log-normal distribution, and B_N is the resource blocks allocated to the users in the 5G network. The transmission power of SBS has a similar definition.

B. Service Model

Currently, the operators' approach is to deploy SBS to certain areas where user traffic is high. Therefore, other areas that request fewer users may be covered only by MBS. When the user i generates a request for the content m, the SBS n that the user contacts would cache the content. If there are a large number of requests for the same content m in other SBS coverage areas, coordinated the request information of the entire network, the strategy would tend to multicast content m directly by the MBS, instead of delivering the content to a certain SBS via the MBS, then multicasting by the SBS.

Consider the general case of MBS and SBS multicast. For SBS n multicast transmission consider that, at time slot t, a request for the content m has been generated in the area covered by the SBS n, and has been cached locally. For MBS multicast transmission consider that in the relevant SBS coverage area, the content m requested by the user in the current time slot t has not been cached.

The multicast caching strategy of all BSs in 5G network depends on how to minimize the energy consumption of the entire network. Use n^* to indicate the coverage area of a BS n where the BS needs the highest transmit power. Then, at time slot t, the BS multicast energy consumption is as in eq. (6):

$$u_{n,m} = P_{n^*} = \max_{n \in \mathcal{N}} P_n, \forall n \in \mathcal{N}. \tag{6}$$

The energy consumption for SBS n to multicast cached content to local users is generally less than the energy consumption of MBS multicast, as eq. (7) indicates.

$$u_{n,m} \leq u_{0,m}, \forall n \in \mathcal{N}^+. \tag{7}$$

Similarly, $v_{n,m}$ is used to represent the energy consumption of the SBS n cache content m. The energy required for the SBS cache is related to the CPU size of the SBS, and is also related to other hardware structures of the SBS. Finally, $w_{0,m} \geq 0$ indicates the energy consumption when the MBS delivers content through the backhaul link. Here, it is necessary to consider the content request change of each SBS in each time slot. It is assumed that in each time slot, SBS n can buffer the content of the s, and when MBS delivers content to SBS, it will consider the state of current cached content of

SBS n. If there are already d contents in the buffer, at time slot t, SBS n will not request the same content from the MBS again, which is mathematically expressed as in eq. (8).

$$w_{0,m} = \sum_{n=1}^{M} (s-d) P_{bh}(n), \forall n \in \mathcal{N}^{+}, \tag{8}$$

where $P_{bh}(n)$ represents the power when the MBS sends a content to the SBS n through the backhaul link.

C. Request Queue Model

In each time slot, the user submits a content request to the MBS or local SBS. The number of requested content arriving at the BS in the time slot t is $\alpha_{n,\ m}$, thus the content request sequence at time slot t can be expressed as in eq. (9).

$$\alpha_{n,m}(t) = \left\{0, 1, 2, \ldots, \alpha_{n,m}^{max}\right\}. \tag{9}$$

Since the users between the BSs do not overlap and the request probabilities are the same, the request sequence of each BS can be considered to be independent and identically distributed (i.i.d). In order to record pending requests, the queue $Q_{n,m}(t)$ associated with the BS and the content m is modeled below. In theory, if the content m is multicast by the MBS, it means that at time slot t, all pending requests in the queue $Q_{0,\ m}(t)$ will be satisfied. Here we find a factor that is easily overlooked: there are usually some less popular contents. These contents are requested less frequently, that is to say, being unsatisfied for a long time. In order that all the requests would not to be ignored, we add a parameter β related to number of times to update the rule of the pending request queue. Therefore, the pending request queue for MBS $(n = 0)$ is updated as in eq. (10).

$$\begin{aligned} Q_{0,m}(t+1) = &\left(1 - x_{0,m}(t)\right) Q_{0,m}(t) + \alpha_{0,m}(t+1) \\ &+ \beta, \end{aligned} \tag{10}$$

where $x_{0,m}(t) = 1$ indicates the MBS multicast scheduling content m, and $x_{0,m}(t) = 0$ shows that the MBS has no multicast content m at time slot t. β is a fixed constant. Similarly, the SBS $(n \in \mathcal{N}^{+})$ pending request queue is dynamic as follows:

$$\begin{aligned} Q_{n,m}(t+1) = &\left\lceil 1 - x_{n,m}(t) - x_{0,m}(t) \right\rceil Q_{n,m}(t) + \\ &\alpha_{n,m}(t+1) + \beta, \end{aligned} \tag{11}$$

where $\lceil \cdot \rceil$ is defined as

$$\lceil x \rceil \overset{def}{=} \begin{cases} x, \ if \ x \geq 0 \\ 0, \ if \ x < 0 \end{cases}. \tag{12}$$

However, the reality is that the resources of each base station are limited. Suppose $R_{n,m}$ is the upper limit value of the user request satisfied by the BS, that is, the number of user pending requests that each BS can satisfy through multicast in each time slot is limited. Therefore, the formula (10) is updated to

$$Q_{0,m}(t+1) = \left\lceil Q_{0,m}(t) - r_{0,m}(t) \right\rceil + \alpha_{0,m}(t+1) + \beta, \tag{13}$$

where

$$r_{0,m}(t) = \begin{cases} R_{0,m}, x_{0,m}(t) = 1 \\ 0, \ x_{0,m}(t) = 0 \end{cases}.$$

Similarly, formula (11) is updated to eq. (14).

$$Q_{n,m}(t+1) = \left\lceil Q_{n,m}(t) - r_{n,m}(t) \right\rceil + \alpha_{n,m}(t+1) + \beta, \tag{14}$$

where

$$r_{n,m}(t) = \begin{cases} R_{n,m}, \ x_{0,m}(t) = 1 \ or \ x_{n,m}(t) = 1 \\ 0, \quad otherwise \end{cases}.$$

In the following sections, the problem with the pending request queue dynamics in this format will be considered. In addition, $Q_{n,m}(t)$ should have an upper limit, as in eq. (15).

$$\sum_{T} Q_{n,m}(t) \leq Q_{max} \tag{15}$$

D. Problem Formulation

Since the SBS n multicast policy depends on its cached content, only two binary matrices x, y are defined as optimization variables, representing the cache and multicast policies, respectively. The value of $x_{n,m}$ indicates whether the content m is stored in the cache of the SBS ($x_{n,m} = 1$ is yes, $x_{n,m} = 0$ is no). Then in slot t, the SBS n caching policy is expressed as in eq. (16).

$$x \triangleq \left\{x_{n,m}(t) \in \{0,1\} : n \in \mathcal{N}^{+}, m \in \mathcal{M}, t \in \mathcal{T}\right\} \tag{16}$$

The value of $y_{n,m}$ indicates whether the BS performs multicast transmission ($y_{n,m} = 1$ is yes, $y_{n,m} = 0$ is no). Then, within slot t, the multicast policies of all BSs are expressed as in eq. (17):

$$y \triangleq \{y_{n,m}(t) \in \{0,1\} : n \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}\} \tag{17}$$

Obviously, the operations of BS scheduling content m consume energy. In time slot t, the energy consumption can be specifically divided into three parts: the BS multicast energy consumption u, the SBS cache energy consumption v, and the energy consumption w on the MBS backhaul link, i.e.

$$u \triangleq \{u_{n,m}(t) : n \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}\}, \tag{18}$$

$$v \triangleq \left\{v_{n,m}(t) : n \in \mathcal{N}^{+}, m \in \mathcal{M}, t \in \mathcal{T}\right\}, \tag{19}$$

$$w \triangleq \{w_{0,m}(t) : m \in \mathcal{M}, t \in \mathcal{T}\}. \tag{20}$$

Therefore, BS n multicast energy consumption is:

$$C_1(t) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \mathbb{E}\left\{y_{n,m}(t)\, u_{n,m}(t)\right\}. \tag{21}$$

SBS cache energy consumption is expressed in eq. (22).

$$C_2(t) = \sum_{n \in \mathcal{N}^{+}} \sum_{m \in \mathcal{M}} \mathbb{E}\left\{x_{n,m}(t)\, v_{n,m}(t)\right\} \tag{22}$$

The energy consumption of content delivered through backhaul link by MBS is as in eq. (23):

$$C_3(t) = \sum_{m \in \mathcal{M}} \mathbb{E}\left\{(s-d) w_{0,m}(t)\right\}, \tag{23}$$

where $\mathbb{E}\{\varphi(t)\}$ shows the time average of $\varphi(t)$. In summary, the total energy consumption is expressed in eq. (24).

$$J(t) = C_1(t) + C_2(t) + C_3(t). \tag{24}$$

Our optimization goal is to determine the strategy of BS cooperative scheduling in 5G networks, aiming to minimize the average energy consumption under the constraints of certain delay conditions.

It is worth noting that we minimize energy consumption on the premise that the pending request queue is stable, which is determined by the number of content requests by users in the entire system against the actions of MBS and SBSs. However, in order to better focus the problem on MBS and SBS scheduling strategies, we design the optimization goal as the sum of the total energy consumption and the number of outstanding requests in the queues. In section V we will further show the performance relationship between energy consumption and throughput. Therefore, *Multicast Aware Caching Scheduling Problem (MACSP)* is described as follows:

Minimize:

$$\frac{1}{T} \sum_{T=0}^{T-1} \sum_{n \in \mathcal{N}, m \in \mathcal{M}} J(t) + Q_{n,m}(t) \tag{25}$$

Subject to:

$$\sum_{m \in \mathcal{M}} y_{0,m}(t) \leq 1 \tag{26a}$$

$$\sum_{n \in \mathcal{N}^+} \sum_{m \in \mathcal{M}} y_{0,m}(t) \leq s \tag{26b}$$

$$y_{n,m}(t) \leq x_{n,m}(t), \forall n \in \mathcal{N}^+ \tag{26c}$$

$$\sum_{T} Q_{n,m}(t) \leq Q_{max} \tag{26d}$$

$$\sum_{m \in \mathcal{M}} x_{n,m}(t) \leq S_n, \ \forall n \in \mathcal{N}^+ \tag{26e}$$

$$x_{n,m}(t) \in \{0,1\}, \ n \in \mathcal{N}^+, m \in \mathcal{M}, t \in \mathcal{T} \tag{26f}$$

$$y_{n,m}(t) \in \{0,1\}, n \in n \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T} \tag{26g}$$

The constraint (26a) expresses that at time slot t, the MBS can only multicast at most one content. The constraint (26b) means that at time slot t, the SBS can cache up to s contents. Constraint (26c) ensures that the content of the SBS n multicast has been cached. Constraint (26d) indicates that all queues are in the normal range to ensure limited accumulation of pending requests. Constraint (26e) shows that the cache space has an upper limit. Constraints (26f)-(26g) represent the discrete properties of the two optimized variables.

IV. Distributed Policy Gradient Algorithm

The MACSP problem is actually trying to find the best balance between the opposing goals to have minimum energy consumption and minimum pending request queue size. From the above optimization problem (25), it can be seen that, unlike traditional optimization problems, we need to find the optimal strategy for MBS and each SBS in a period of time T, rather than simply solving the optimal value in each time slot (like greedy algorithm, which would likely not find a global optimal solution). The MACSP problem solving involves the time stationary of the sequence, which obviously is a challenge.

Unlike supervised learning, reinforcement learning does not rely on the prepared data for training. It only has a reward value, and this reward value is different from the output value of supervised learning. It is not given in advance, but is given based on the delay of the action. Therefore we consider using reinforcement learning to solve this problem. In this section, we use Markov Decision Process to simplify our reinforcement learning model, and we propose the *Distributed Policy Gradient (DPG) Algorithm* based on Deep Deterministic Policy Gradient (DDPG) training and Deep Q-Learning (DQN) algorithm. This algorithm is used to solve the MACSP problem, noting that a coupling problem is creatively solved separately.

A. Markov Decision Process

A Markov Decision Process involves a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is a set of state matrices named state space. $\mathcal{A}$ is a set of behavioral matrices, we call it action space. The state transition probability refers to the probability that the time slot t to the next time slot state becomes s, which expressed as $\mathcal{P}$. The reward function is a reward that is received immediately after the action $a(t)$ transitions from the state $s(t)$ to the state $s(t+1)$, which represented by $\mathcal{R}$. γ is a decay factor to adjust learning rate, $\gamma \in [0,1]$.

1) The State Space: We assume that the system state changes at an independent time. The time that the system remains in its current state until the next state is called the phase. Since a matrix representation is advantageous for integrating the deep learning framework, we focus on the problem given the system state at slot $t \in T$, and the response state of MBS and SBSs represented by a matrix $\mathcal{S} = [s_{n,m}]$, where $s_{n,m}$ is the number of content m requests by users in SBS n. We consider two situations: when the content m is not requested, we would set a special value for $s_{n,m}$; correspondingly, if the content m is requested, $s_{n,m}$ will obey the rule from eq. (27):

$$s_{n,m}(t) = \frac{Q_{n,m}(t)}{\sum_{n \in \mathcal{N}^*} Q_{n,m}(t)}, \tag{27}$$

where $Q_{n,m}$ is the length of pending request queue of SBS n to content m. The present state set of the system can be expressed as in eq. (28):

$$\mathcal{S} = \{(s_{n,m}, \Delta t) \,|\, s_{n,m} \in \mathbb{S}, \Delta t \in \mathbb{T}\}, \tag{28}$$

where $\mathbb{S}$ is the set of entire random combination, $\mathbb{T}$ is the set of all action time gap of the agent.

2) The Action Space: When the system state is S, the set of possible actions A is a combination trained by the reinforcement learning algorithm. Due to the huge variety of content in the 5G network, the content combination of SBS cache and multicast is also very large, so the action of the system's multicast cache cannot be simply regarded as discrete behavior, but should be regarded as a continuous variable. The traditional DDPG algorithm is to use a deeper network structure, plus a strategy gradient algorithm, to randomly select actions in a continuous action space according to the learned strategy (action distribution). The role of Deterministic is to help the policy gradient not to randomly chooses to output only one action value. In other words, the policy output is the action, i.e. $\pi(s) : \mathcal{S} \rightarrow \mathcal{A}$, and a policy π is a probability distribution over actions in given states.

$$\pi\left(a|s\right) = P\left[A^t = a|S^t = s\right]. \tag{29}$$

Therefore, each BS requests a different content m to have a corresponding action value, and the action matrix is represented by $\mathcal{A}$, where a represents the value trained by our algorithm. Define the action space as all possible combinations of BS cooperative behavior, i.e.

$$\mathcal{A} = \bigcup A^t, \tag{30}$$

where $\mathcal{A}$ is a set of contents allocation solutions.

3) State Transition Probability: The state transition probability from state $S_j(t)$ to state $S_k(t)$ is given by eq. (31):

$$p_{j,k} = p\left(A^t, \Delta t\right), \tag{31}$$

where $p(A^t, \theta t)$ refers to the system state transition rate divided by the system state transition probability from state j to state k divided by the overall state transition rate from state j. Specifically, the state transition rate from state j to state k is the probability of occurrence of state j transition to k, and the total state transition rate is the frequency of occurrence of all possible events. Although the expression of the state transition matrix probability is given here, the mapping relationship cannot be directly used to solve the problem because it has no application value in the real situation.

4) Reward Function: The reward function defines the goal in the reinforcement learning problem, which maps each perceived state to a single value $R : \mathcal{S} \times \mathcal{A} \rightarrow R$, indicating the intrinsic demand of the state. In the MACSP problem, the goal is to select the appropriate operation for each BS in different states to optimize the overall performance of the network and the quality of user service. We roughly divide the reward into two parts, SBS behavior cost (SBS for caching and small-scale multicast energy consumption) and MBS behavior cost (MBS multicast and energy consumption on its backhaul link). Energy consumption issues and queue constraints have been highlighted in Section III, so the reward function is expressed as in eq. (32):

$$R(t) = -Q_{n,m}(t) - J(t), \tag{32}$$

where $Q_{n,m}$ indicates the pending request queue, and $J(t)$ gather the whole energy consumption.

B. Algorithm Design

Different from the greedy strategy, our distributed algorithm based on DDPG and DQN can complete the training space of the Markov process, so as to make decisions.

Since the SBS can multicast and cache multiple pieces of content in the time slot t, assuming that each time slot SBS n multicasts c pieces of contents, then the reward function obtained by the state corresponding to each action of the actor has a specific expression, i.e. formula (32). The Policy Gradient (PG) method employs a random strategy, and if the goal is to perform the action in the current state, we need to sample the probability distribution of the optimal strategy to train the value of the desired action space. Therefore, we adopt a deterministic strategy based on PG, and determine an action based on the behavior directly through the function μ. This is an optimal behavior strategy expressed as in eq. (34).

$$a(t) = \mu\left[s(t)|\theta\right]. \tag{33}$$

This deterministic strategy μ is used to select the action, where θ is the parameter of the strategic network that produces deterministic actions.

Use the policy network μ to act as an actor, and use the value network to fit the (s, a) function to play the role of critic, so the objective function of DDPG can be defined as

$$F(\mu_\theta) = \mathbb{E}_{s\sim s^\mu}\left[r(s, \mu_\theta(s))\right], \tag{34}$$

where $r(\cdot)$ denotes the reward function of state s. At this point, the Q function is expressed as the expected reward value for choosing actions under a deterministic strategy μ. Here we use a Q network to fit the Q function.

$$Q^\mu\left(s^t, a^t\right) = \mathbb{E}\left[r\left(s^t, a^t\right) + \gamma Q^\mu\left(s^{t+1}, a^{t+1}\right)\right] \tag{35}$$

The formula from eq. (36) is used to evaluate the quality of strategy μ.

$$J_\beta(\mu) = E_{s\sim s^\beta}\left[Q^\mu(s, \mu(s))\right], \tag{36}$$

where β is the random noise we introduced for the decision mechanism of the action, and it obeys the Uhlenbeck-Ornstein stochastic process.

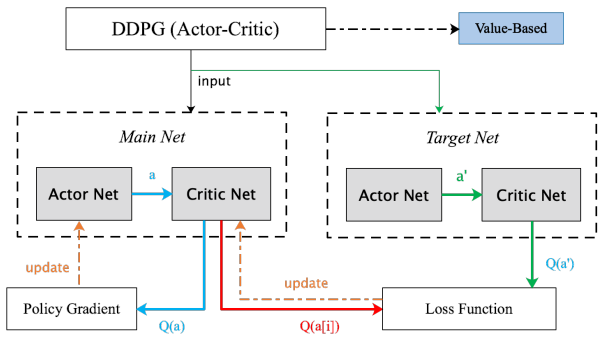


Fig. 2: Network structure of DDPG

Algorithm 1: Step 1 for single SBS training

Initialize: actor network $\mu\left(s|\theta^{\mu}\right)$, critic network $Q\left(s,a|\theta^{Q}\right)$, target network and replay buffer.
for *each episode in range* **do**
  Initialize noise β as an OU random process for later exploration;
  for *every steps in range* **do**
    Select action using $a_t = \mu\left(s|\theta^{\mu}\right) + \beta$;
    Perform actions in the environment;
    reward = (the cost of multicast content as a percentage of all content in the pending queue) + (bh_cost + $cache\_cost$ + $multicast\_cost$)$/C$;
  Storage sample (s_t, a, r, s_{t+1}) into reply buffer;
  Set $y_i = r_i + \gamma Q\left(s_{i+1}, \mu\left(s_{i+1}|\theta^{\mu}\right)|\theta^{Q}\right)$;
  loss = $\frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i|\theta^{Q}\right)\right)^2$;
  Update critic by minimizing the loss function;
  Update the actor policy using the sampled policy gradient;
  Update the target networks;

Related to Deep Deterministic Policy Gradient (DDPG), the DPG algorithm uses the deterministic strategy μ to select action a_t, which is consistent with the requirements of MACSP, where μ is the parameter that produces the deterministic action strategy. We use the strategy μ to act as the actor required for training, and use the critic net to fit the $Q(a)$ function to act as a critic. Because the structure of Deep Q-Learning (DQN) is referenced, there is one more target network in Fig. 2. Lillicrap et al. gave a detailed proof of the network update using gradient descent in the DDPG algorithm [32].

In our actor-critic framework, the action space of the critic net is simplified, which reduces the computational complexity of the algorithm and thus converges better. The proposed DPG algorithm mainly includes two stages: first, an SBS agent would be trained separately and secondly, the SBS agent and MBS would be integrated to train together. Therefore the algorithm is divided into two parts. In details, first we initialize the actor and critic network and create the target network. Then, the value of A is continuously updated during training using the reinforcement learning method to obtain the output value of the action space in the next iteration. Finally, all the queues are updated at the current time to get a new target network. This iteration is summarized in Algorithm 1.

We assume that the number of multicasts of MBS in a slot should be very small, so the MBS multicast action space is relatively large. Considering that DDPG has a poor training effect on discrete action spaces, DQN is used for final integration training. Discrete values can be used to indicate which content is selected from existing content after all. The integration stage is described in Algorithm 2.

We performed extensive numerical simulations presented in the next section, and the results show that the solution performs better in real settings than when employing alternative approaches.

V. Simulation-based Testing

The simulation results will be presented to show the feasibility of the proposed algorithm. Besides, we also introduced a trivial greedy algorithm in comparison to show the effectiveness of our algorithm.

A. Environment Setup

In order to solve the MACSP problem with the proposed deep reinforcement learning approach, there was a need to build the environment for the simulation scenario. We consider there are 4 SBS and 1 MBS in the environment and each SBS can cache up to 2 content entries at any time. We also consider the number of content types equal to 5. In each time slot, 15 user requests are sent to each SBS. Without losing any generality, we suppose that each SBS can multicast 2 pieces of content to satisfy user requests. In the meanwhile, each MBS can multicast 1 piece of content to satisfy the remaining requests. Each SBS has a delay queue which contains requests to be handled. We assume that each request has a delay time attribute which indicates its priority. To retain the crucial property of user requests, we use the Zipf distribution to generate user requests. The Zipf distribution assumes that the content has associated a popularity as in eq. (37):

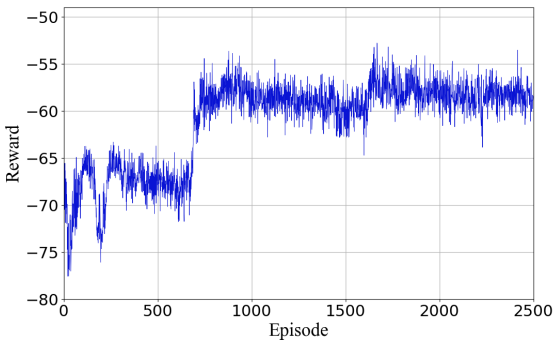
$$f\left(k,a\right) = \frac{1}{\zeta(a)k^{a}}, \tag{37}$$

where a has value of 1.2 in our simulations. The simulation parameters are listed in Table I for convenience.

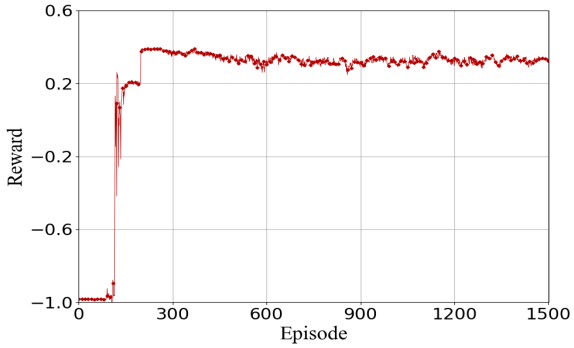
For simplicity, at the beginning of every episode, each SBS cache content should also follow the Zipf distribution. To demonstrate the efficiency of the reinforcement learning approach, a basic greedy multicast method and a trivial random

Algorithm 2: Step 2 for all BSs Training (integrating MBS base on step 1)

Initialize: DQN's reply memory and the SBS agent that has been trained using the DDPG algorithm, randomize the Q function of DQN as well. **for** *each episode in range* **do**
  Initialize $s\_observation$ of every SBS;
  The agent takes actions to update status according to $s\_observation$ of SBS;
  if *get the current status* **then**
    select a random action with probability ε;
  else
    $action = \max_a Q^{*}\left(s_t, a, \theta\right)$;
  reward = (the cost of multicast content as a percentage of all content in the pending queue) + (bh_cost + $cache\_cost$ + $multicast\_cost$)$/C$;
  Storage state space (s_t, a, r, s_{t+1}) in reply memory;
  Get samples in random minibatch of replay memory;
  Update $y_j = r_j + \gamma \max_a Q\left(s_{j+1}, a, \theta\right)$;
  Derived the gradient descent value of y;

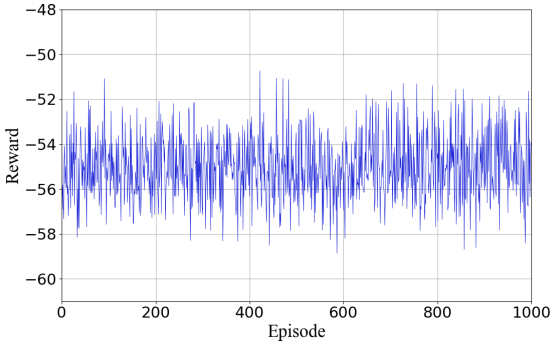


(a) Algorithm 1 for Single SBS Training

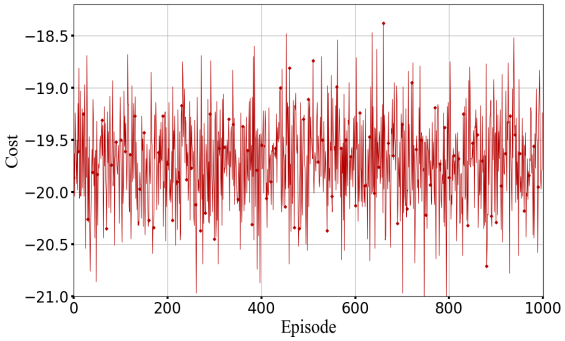


(b) Algorithm 2 for all BSs Training

Fig. 3: Training Results of DPG



(a) Test Result of Single SBS Agent



(b) Test Result after Two Stages

Fig. 4: Testing Results of DPG

TABLE I: SIMULATION PARAMETERS

Parameter	Value
Number of SBS	4
Number of MBS	1
Cache size of each SBS	2
Number of content types	5
User request number per slot	15
User request popularity distribution	ZipF with a = 1.2

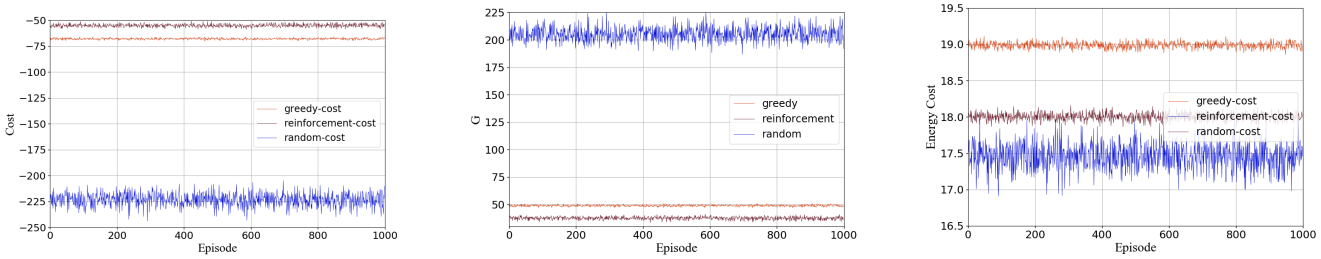
multicast method are used for comparison. The greedy method follows the rule that each SBS selects the content which delays the longest time without considering the energy cost of the entire network system, after all SBS multicast their cached contents to satisfy user request, the MBS selects the remaining content among all SBS which have the longest delay time. Meanwhile, the random method follows the rule that each SBS randomly select contents from its delay queue, and MBS follows the same rule. By using the above greedy rule, the algorithm can obtain a relative short delay queue but the total energy cost is not considered.

The computational complexity of reinforcement learning algorithms is determined by the state space and action space of the problem and size of the network. In our approach, we have divided the algorithm into two stages. First, a single SBS should train its own DDPG agent without considering the MBS's multicast process. The DDPG agent can achieve a relative good SBS multicast strategy having constraint the length of delay queue. Considering the efficiency of the training process, we decided that the action space for single SBS is continuous, represented by a 1-D array with length equal with the number of content items. For each element in the action space the entry in this array indicates whether the content it represents is multicast or not. By default, if the element has a value greater than zero, then SBS should multicast the content. The observation space for the single SBS case is also continuous, represented by a 1-D array indicating the status of the SBS delay queue. Secondly, all SBS and MBS are put together to train a DQN agent to control the MBS multicast process. The action space and observation space in this stage is similar to that in the first stage. We use the default parameters indicated in [34] for the first stage training, and those employed in [35] for the second stage training.

B. Discussion of Results

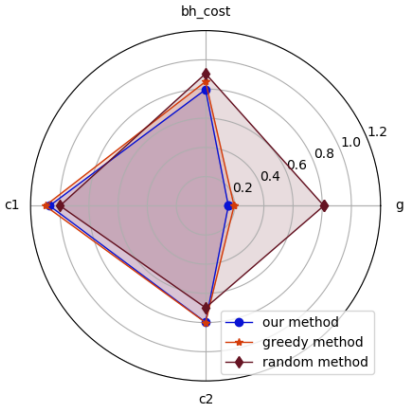
Next the effectiveness of the two stages of the DPG algorithm is assessed in comparison with the results of the greedy algorithm and random algorithm when solving MACSP. The simulation parameters listed in Table I are used.

Fig. 3 illustrates the convergence effect of the DPG algorithm in our problem. As shown in Fig. 3(a), after around 700 episodes, the agent reaches a relative high reward and has the ability to make reasonable decisions with a deep understanding of the environment. Fig. 3(b) presents how by integrating the training process of all BSs and observing the reward value of each step, from about 170 episodes, the reward value is in a stable state. This is as our DPG algorithm chooses a relative

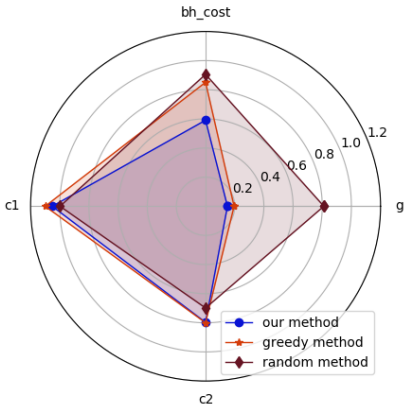


(a) The Value of Cost (b) Waiting Time of the Queue of Requests (c) Energy Consumption of the Entire Network

Fig. 5: Test Results of DPG



(a) a=1.2 of ZipF



(b) a=1.4 of ZipF

Fig. 6: The comparative performance of the three methods in terms of four aspects: c1 denotes the energy consumption of MBS multicasting, c2 indicates the energy consumption of all BSs caching, bh_cost means the energy consumption in backhaul links, and g represents the queue delay.

good action to perform based on current system state, which means that it always tries to satisfy the constraint and keep the cost low.

Fig. 4 shows the test results of DDPG Agent trained by step 1 algorithm and DQN Agent trained by step 2 algorithm for all BSs. The test results of a single SBS training agent are shown in Fig. 4(a). In addition, after joining MBS multicast, the original algorithm (i.e. Step 1) that only includes SBSs multicast can be optimized. Compared with the result of directly superimposing SBS, our DPG algorithm can obtain a lower optimization target cost, which fluctuates around -19.7 in Fig. 4(b). The overall cost value is stable, and the cost value is relatively low. Specifically, in the step 1 algorithm, each SBS makes a relative good decision based on its own state. In step 2, as MBS holds the entire system's state, it takes good actions to address all SBS unsatisfied requests, which reduces the costs.

From these two plots, it can be noted that the proposed DPG algorithm obtains more stable test results for the probability distribution of different requests, and its cost is small. It can also be suggested that Step 2 of DPG (i.e. Algorithm 2) can help MBS and SBS cooperate to perform multicasting.

Fig. 5 shows the performance comparison of our method (the DPG algorithm) with the greedy and random algorithms. It can be seen that the DPG algorithm has a lower cost than the greedy algorithm in Fig.5(a). On the other hand, the results of the greedy and reinforcement learning algorithms are significantly better than that of the random algorithms. Fig. 5(b) shows the total waiting time of each SBS queue under the three algorithms. Although the greedy strategy is to select the content with the longest waiting time for multicast in each step, the sum of the final waiting time is still higher than that of our algorithm. This demonstrates that our algorithm optimizes the queue waiting time in the context of the MACSP problem. Fig. 5(c) shows that our algorithm consumes less energy than the random algorithm and greedy algorithm, which demonstrates that the algorithm also optimizes well the energy consumption, successfully solving the problems we have focused on.

The radar charts shown in Fig. 6 reflect the overall performance of the DPG algorithm related to various aspects. The four indicators in the radar chart are connected into an irregular quadrilateral. From a single point of view, the energy consumption of our algorithm ($c1$ and $c2$ in blue) is slightly larger than for the random algorithm. This is because there is more distributed content under the DPG algorithm, but the delay (g in blue) is much smaller than the delay experienced by the random and greedy algorithms. We believe this is a necessary sacrifice for achieving low latency. In order to comprehensively evaluate the combination of multicast and caching, multiple performance parameters should be considered in conjunction. In this context, the area of the radar

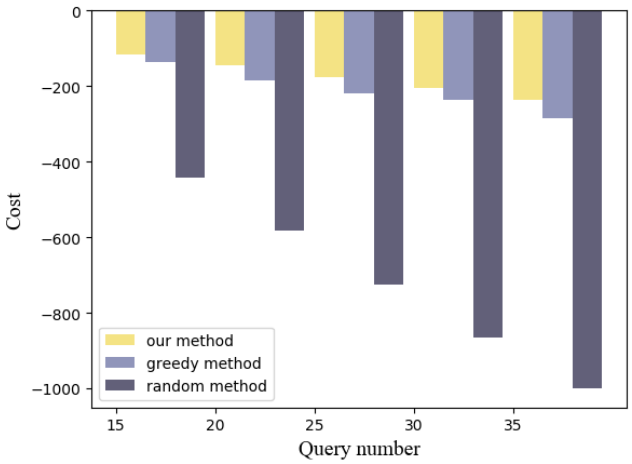


Fig. 7: System performance of three methods

chart illustrates the superiority of our algorithm, that is, the smaller the quadrilateral area, the better overall performance an algorithm can achieve. Additionally, it can be observed in Fig. 6 that the area of the quadrilateral figure surrounded by the blue line is much smaller than the quadrilateral area represented by the other two algorithms. This shows that the overall performance of the proposed algorithm is superior to both greedy and random algorithms. Further, comparing Fig.6(a) with Fig.6(b) by changing the parameter a of ZipF distribution, it can be distinctly seen that the quadrilateral area is different, indicating that our algorithm has better sensitivity.

The relationship between system throughput and energy consumption can be clearly seen from Fig. 7. Under the premise of ensuring a certain throughput, our algorithm has a lower cost than the other two methods. This gap is more obvious in the case of large throughput (that is, for longer queue of pending requests).

VI. Conclusions

This paper focuses on the joint optimal multicast scheduling in cache-enabled wireless heterogeneous networks. We describe the issue as Multicast-Aware Caching Scheduling Problem (MACSP) and, considering its complexity, we employ a Markov Decision Process which benefits from training using reinforcement learning to find the optimal solution. We also propose a Distributed Policy Gradient algorithm (DPG) with similar performance with existing solutions, but lower complexity. Extensive simulation-based experimental testing has showed that our algorithm successfully solves this complex problem, and that outperforms alternative approaches in terms of performance and energy saving.

References

[1] Ericsson, "Mobility report: On the pulse of networked society," June 2015. [Online]. Available: http://www.ericsson.com/mobility-report

[2] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting Caching and Multicast for 5G Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2995–3007, April 2016.

[3] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 621–634, June 2018.

[4] C. Xu, S. Jia, L. Zhong, H. Zhang, and G. Muntean, "Ant-Inspired Mini-Community-Based Solution for Video-On-Demand Services in Wireless Mobile Networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, June 2014.

[5] C. Xu, M. Wang, X. Chen, L. Zhong, and L. A. Grieco, "Optimal Information Centric Caching in 5G Device-to-Device Communications," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2114–2126, Sep. 2018.

[6] G. Araniti, P. Scopelliti, G. Muntean, and A. Iera, "A Hybrid Unicast-Multicast Network Selection for Video Deliveries in Dense Heterogeneous Network Environments," *IEEE Transactions on Broadcasting*, vol. 65, no. 1, pp. 83–93, 2019.

[7] G. Araniti, M. Condoluci, P. Scopelliti, A. Molinaro, and A. Iera, "Multicasting over Emerging 5G Networks: Challenges and Perspectives," *IEEE Network*, vol. 31, no. 2, pp. 80–89, March 2017.

[8] S. Yang, C. Xu, L. Zhong, J. Shen, and G. Muntean, "A QoE-Driven Multicast Strategy With Segment Routing - A Novel Multimedia Traffic Engineering Paradigm," *IEEE Transactions on Broadcasting*, vol. 66, no. 1, pp. 34–46, March 2020.

[9] "3rd Generation Partnership Project (3GPP)," 2016. [Online]. Available: http://www.3gpp.org/speci?cations/releases/71-release-9

[10] T. Lohmar, M. Slssingar, V. Kenehan, and S. Puustinen, "Delivering Content with LTE Broadcast," *Ericsson Review*, Feb 2013.

[11] "LTE Broadcast - A Revenue Enabler in the Mobile Media Era," *Qualcomm White Paper*, February 2013. [Online]. Available: https://www.qualcomm.com/media/documents/files/lte-broadcast-a-revenue-enabler-in-the-mobile-media-era.pdf

[12] A. De La Fuente, G. Femenias, F. Riera-Palou, and A. Garcia Armada, "Subband CQI Feedback-Based Multicast Resource Allocation in MIMO-OFDMA Networks," *IEEE Transactions on Broadcasting*, vol. 64, no. 4, pp. 846–864, Dec 2018.

[13] J. Lee, Y. Kim, Y. Kwak, J. Zhang, A. Papasakellariou, T. Novlan, C. Sun, and Y. Li, "LTE-advanced in 3GPP Rel -13/14: an evolution toward 5G," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 36–42, March 2016.

[14] L. Zhang, Y. Wu, G. K. Walker, W. Li, K. Salehian, and A. Florea, "Improving LTE e MBMS With Extended OFDM Parameters and Layered-Division-Multiplexing," *IEEE Transactions on Broadcasting*, vol. 63, no. 1, pp. 32–47, March 2017.

[15] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid Content Caching in 5G Wireless Networks: Cloud Versus Edge Caching," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3030–3045, May 2018.

[16] D. Liu and C. Yang, "Energy Efficiency of Downlink Networks With Caching at Base Stations," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 907–922, April 2016.

[17] M. Europe, "Altobridge debuts intel-based network edge small cells caching solution," *London, UK, Jun*, 2013.

[18] L. Reading, "Nsn adds chinacache smarts to liquid applications," *New York, NY, USA*, 2014.

[19] Y. Cui, Z. Wang, Y. Yang, F. Yang, L. Ding, and L. Qian, "Joint and Competitive Caching Designs in Large-Scale Multi-Tier Wireless Multicasting Networks," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 3108–3121, July 2018.

[20] B. Dai, Y. Liu, and W. Yu, "Optimized Base-Station Cache Allocation for Cloud Radio Access Network With Multicast Backhaul," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1737–1750, Aug 2018.

[21] B. Zhou, Y. Cui, and M. Tao, "Optimal Dynamic Multicast Scheduling for Cache-Enabled Content-Centric Wireless Networks," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2956–2970, July 2017.

[22] B. Zhou and Y. Cui, "Stochastic Content-Centric Multicast Scheduling for Cache-Enabled Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6284–6297, Sep. 2016.

[23] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Cache-Aware Multicast Beamforming Design for Multicell Multigroup Multicast," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11 681–11 693, Dec 2018.

[24] S. Nikolaou, R. Van Renesse, and N. Schiper, "Proactive Cache Placement on Cooperative Client Caches for Online Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1174–1186, April 2016.

[25] J. Zhang and P. Elia, "Fundamental Limits of Cache-Aided Wireless BC: Interplay of Coded-Caching and CSIT Feedback," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3142–3160, May 2017.

[26] T. Zhang, X. Xu, Le Zhou, X. Jiang, and J. Loo, "Cache Space Efficient Caching Scheme for Content-Centric Mobile Ad Hoc Networks," *IEEE Systems Journal*, vol. 13, no. 1, pp. 530–541, March 2019.

[27] Y. Zhang, Q. Wei, C. Chen, M. Xue, X. Yuan, and C. Wang, "Dynamic Scheduling with Service Curve for QoS Guarantee of Large-Scale Cloud Storage," *IEEE Transactions on Computers*, vol. 67, no. 4, pp. 457–468, April 2018.

[28] X. Zhang, T. Lv, Y. Ren, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Economical Caching for Scalable Videos in Cache-Enabled Heterogeneous Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1608–1621, July 2019.

[29] G. H. Sim, J. Widmer, and B. Rengarajan, "Opportunistic Finite Horizon Multicasting of Erasure-Coded Data," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 705–718, March 2016.

[30] J. Rückert, J. Blendin, R. Hark, and D. Hausheer, "Flexible, Efficient, and Scalable Software-defined over-the-top Multicast for ISP Environments with DYNSDM," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 754–767, 2016.

[31] H. Soni, W. Dabbous, T. Turletti, and H. Asaeda, "NFV-Based Scalable Guaranteed-Bandwidth Multicast Service for Software Defined ISP Networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1157–1170, Dec 2017.

[32] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-Enabled Multicasting in SDNs," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2052–2070, March 2019.

[33] X. Li, Y. Tian, G. Ledwich, Y. Mishra, and C. Zhou, "Minimizing Multicast Routing Delay in Multiple Multicast Trees With Shared Links for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5427–5435, Sep. 2019.

[34] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[35] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ser. ICML'16, 2016, p. 1995–2003.

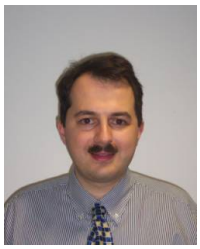
Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is currently an Associate Professor with the Information Engineering College, Capital Normal University, Beijing, China. Her research interests include communication networks,computer system and architecture,and mobile Internet technology.

Changqiao Xu (SM'15) received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences (ISCAS) in Jan. 2009. He was an Assistant Research Fellow and R&D Project Manager in ISCAS from 2002 to 2007. He was a researcher at Athlone Institute of Technology and Joint PhD at Dublin City University, Ireland during 2007-2009. He joined Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in Dec. 2009. Currently, he is a full Professor with the State Key Laboratory of Networking and Switching Technology, and Director of the Next Generation Internet Technology Research Center at BUPT. His research interests include Future Internet Technology, Mobile Networking, Multimedia Communications, and Network Security. He has published over 160 technical papers in prestigious international journals and conferences. He has served a number of international conferences and workshops as a Co-Chair and Technical Program Committee member. He is currently serving as the Editor-in-Chief of Transactions on Emerging Telecommunications Technologies (Wiley). He is Senior member of IEEE.

Jiewei Chen is a master candidate in Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include wireless network, edge computing and federated machine learning.

Weiqi Yan received the B.E. degree in digital media technology from Communication University of China, Beijing in 2019. Now he is pursuing a Master's degree of Computer technology in Beijing University of Posts and Telecommunication, China.

Shujie Yang received the Ph.D. degree in Institute of Network Technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently a Lecturer with State Key Laboratory of Networking and Switching Technology, Beijing, China. His major research interests are in the areas of wireless communications, and wireless networking.



Gabriel-Miro Muntean (SM'17) is an Associate Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Laboratory. He has published over 350 papers in top-level international journals and conferences, authored 4 books and 19 book chapters, and edited 6 additional books. He has supervised to completion 22 Ph.D. students and has mentored ten post-doctoral researchers. His research interests include quality, performance, and energy saving issues related to multimedia and multiple sensorial media delivery, technology enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and chair and reviewer for important international journals, conferences, and funding agencies. He is a Senior Member of IEEE Broadcast Technology Society.