Energy Minimization in D2D-Assisted Cache-Enabled Internet of Things: A Deep Reinforcement Learning Approach

Jie Tang, *Senior Member, IEEE*, Hengbin Tang, Xiuyin Zhang, *Senior Member, IEEE*,
Kanapathippillai Cumanan, *Senior Member, IEEE*, Gaojie Chen, *Senior Member, IEEE*,
Kai-Kit Wong, *Fellow, IEEE*, and Jonathon Chambers, *Fellow, IEEE*

***Abstract*—Mobile edge caching (MEC) and device to device (D2D) communications are two potential technologies to resolve traffic overload problems in internet of things (IoT). Previous works usually investigate them separately with MEC for traffic offloading and D2D for information transmission. In this paper, a joint framework consisting of MEC and cache-enabled D2D communications is proposed to minimize the energy cost of systematic traffic transmission, where file popularity and user preference are the critical criteria for small base stations (SBSs) and user devices, respectively. Under this framework, we propose a novel caching strategy where Markov decision process (MDP) is applied to model the requesting behaviours. A novel scheme based on reinforcement learning (RL) is proposed to reveal the popularity of files as well as users' preference. In particular, Q-learning (QL) algorithm and deep Q-network (DQN) algorithm are respectively applied to user devices and SBS due to different complexities of status. To save the energy cost of systematic traffic transmission, users acquire partial traffic through D2D communications based on the cached contents and user distribution. Taking the memory limits, D2D available files and status changing into consideration, the proposed RL algorithm enables user devices and SBS to prefetch the optimal files while learning, which can reduce the energy cost significantly. Simulation results demonstrate the superior energy saving performance of the proposed RL-based algorithm over other existing methods under various conditions.**

***Index Terms*—Content caching, D2D communications, internet of things (IoT), Q-learning (QL), deep Q-network (DQN).**

I. Introduction

With the advent of the fifth generation (5G) communication era, dramatic increasing in the number of data devices such as smart-phones, internet of things (IoT) devices have emerged and led to an exponential growth in data services. To support those devices and the high volume of the data traffic, 5G will require a paradigm shift that includes very high carrier frequencies with massive bandwidths, extreme base station and device densities and unprecedented numbers of antennas. However, the traditional communication networks is far from sufficient to undertake the traffic demands. This motivates the need to develop new technologies such as millimetre wave, massive multiple-input multiple-output (MIMO), machine-to-machine communications, and they will lead to fundamental changes in 5G and beyond wireless networks [1].

To cater the massive traffic demands and universal high data rate, the idea that offloading the popular traffic to the communication network edges was proposed and has attracted a significant attention recently in both academia and industry [2], [3]. This caching scheme, i.e., mobile edge caching (MEC), proactively fetches the content and caches them at the edge nodes, e.g., small base stations (SBS) and user devices. Since the edge nodes served users directly, MEC resolves the backhaul constraints efficiently, and hence, the transmission latency reduces significantly. In most research work, MEC consists of two core stages, one is the content delivery stage, in which the users' requests will be satisfied. The other is content placement stage, in which the selected content will be placed in the edge nodes [2]. The content placement stage mainly relies on the caching policy, where the storage limits, user preference, caching locations etc. are taken into consideration. To address the policy selection problem, caching strategies with different optimization objectives have been investigated, e.g., the coded caching scheme [3] and the learning based centralized caching scheme [4]. Furthermore, many other caching assisted applications are studied as well, e.g., recommendation policy based on caching content [5], small population content caching policy [6], caching enabled energy efficiency optimization [7].

In addition to the caching scheme, device-to-device (D2D) communication is another key technology to improve system capacity and resolve backhaul congestion. D2D scheme allows the devices to establish direct communication link between devices through bypassing the base station [8]. As a result, both the system capacity and transmission energy saving can be improved significantly. It can be observed that D2D has been recognized as a key technology for 5G and beyond

This paper was presented in part at the IEEE Global Communications Conference (GLOBECOM), Waikoloa, Hawaii, 2019. This work has been supported in part by the National Natural Science Foundation of China under Grant 61971194 and 61601186, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030313383, in part by the Fundamental Research Funds for the Central Universities under Grant 2019JQ08, and in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University (No. 2019D06). *(Corresponding author: Gaojie Chen)*

J. Tang, H. Tang and X. Zhang are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. (e-mail: eejtang@scut.edu.cn; ido1194@outlook.com; zhangxiuyin@scut.edu.cn).

K. Cumanan is with the Department of Electronic Engineering, University of York, United Kingdom. (e-mail: kanapathippillai.cumanan@york.ac.uk).

G. Chen and J. Chambers are with the Department of Engineering, University of Leicester, United Kingdom. (email: gaojie.chen@leicester.ac.uk; jonathon.chambers@leicester.ac.uk).

K.-K Wong is with the Department of Electronic and Electrical Engineering,University College London, London WC1E 7JE, U.K. (e-mail:,kai-kit.wong@ucl.ac.uk).

wireless networks and investigated widely in various scenarios, for example, the mode selection problem of D2D was solved in [8]. In [9], authors propose multiantenna transceiver design and multihop D2D communication to guarantee the reliable transmission and extend the UAV coverage for IoT in disasters, and the results confirm the performance improvement in the throughput and outage probability by the proposed approaches. In [10], the optimal routing was proposed for multi-hop D2D communications. In [11], a D2D-assisted caching strategy is investigate, and a non-parametric estimator is proposed to estimate a optimal $\langle \text{user file} \rangle$ pairs for efficient caching. Moreover, In [12], Wang $et\ al.$ study the computation and traffic offloading in cache-aided device-to-device multicast networks for the content delivery and delay sensitive task offloading services. In [13], a D2D-assisted machine type communication model was explored.

With the great progress of machine learning (ML) recent years, increasing research have adopted ML to solve complicated communication problems, for example, a new deep learning based Non-Orthogonal Multiple Access scheme which can detect the channel characteristics automatically [14], IoT feature extraction and reuse [15], caching file selections [16]. Especially the branch of deep learning is suitable to deal with some non-convex optimization problems. Although there exists a complete set of convex optimization theories, the communication environment nowadays gets more and more complex, it is almost impossible to formulate a pure convex problem in many scenarios. Therefore, more and more communication scientists focus on developing the potential of ML for some tricky communication problems. As an important branch, deep reinforcement learning (RL) has attracted great attention recently [17]–[22]. The authors in [17] investigate the computation offloading problem in blockchain empowered mobile edge computing system, where the deep RL algorithm is applied to the computing offloading decision-making process. Moreover, deep RL is also applied in unmanned aerial vehicle autonomous target searching in a complex disaster scene [18], where the superior ability on dynamic programming of deep RL can be observed. In addition, some well-known deep RL algorithms such as SARSA [19], DQN [20]–[22] are investigated and exploited in practical communication systems. In particular, the DQN algorithm is used for resource allocation in edge computing networks [20], dynamic multichannel access problem in wireless networks [21] and mobile robots path planning problems [22].

A. Prior work

Exposing the popularity of requested files is the primary goal of existing works, where most of the works take the content popularity as the main criteria to decide which files should be cached [23]–[25]. Specifically, an online Pop-Caching scheme has been used to learn the popularity of files to determine which content it should store and which it should evict from the cache in [23]. In [24], the caching and scheduling policies were jointly optimized to maximize successful offloading probability, D2D and caching are adopted simultaneously and the offloading gain are remarkably improved. In [25], the problem of video file caching based on wireless D2D was investigated, in which mobile users were designated as helpers store popular video files and serve other requesting users via D2D localized transmissions. In [26], the authors investigate the outage probability and symbol-error rate for both full duple and full-duplex transmission schemes in multihop networks subject to interference from randomly distributed third-party devices. Differ from [26], in this paper we focus on the minimization of energy cost, where file popularity and user preference are the critical criteria for SBSs and user devices, respectively.

Additionally, an architecture based on distributed caching content in femto-base stations with helper nodes was proposed in [27], in which D2D was applied to distribute the video. Furthermore, in some recent research, ML based scheme is adopted in wireless communications. In [28], a deep reinforcement learning (RL) method for resource sharing and caching problem has been investigated. The deep RL approach was applied to automatically make decision for optimally allocating the resource, and the simulations prove the superior performance of deep RL. However, the learning based caching policies in D2D-assisted IoT remain to be explored and it is worth studying on the joint caching technique based on D2D and ML. Specially, a DQN algorithm is applied on mobile robots path planning [20], where the DQN algorithm trained an action value function for action estimation. In addition, a dynamic multichannel access problem is investigated with DQN in [22], where the problem is modelled as a Markov decision process (MDP) with unknown system dynamics. Particularly, the DQN algorithm can achieve near-optimal performance, and it works better than other algorithms in a more complex situation. In other words, the channel selection decision can be efficiently solved with the help of DQN. In [29], the total system power consumption minimization problem in a cache-enabled mobile network is considered, the authors decouple the optimization task into several subproblems and solve them with the idea of associating the users with the SBS. In our work, we use a deep RL algorithm to solve the file selection problem, and the optimal caching files can be predicted directly.

B. Contributions

A systematic traffic transmission energy cost minimization problem is investigated in this paper. Due to the complexity and randomness of user requests, the energy minimization problem turns out to be non-deterministic polynomial (NP) hard. Hence the corresponding optimal solution are of great computational complexity with conventional approaches. In this case, a joint online scheme based on RL is proposed in this paper. We model the user preference with Zipf distribution rigorously to assist the RL algorithms. The RL algorithms, that extract the underlying file popularities, are taken to predict the next-slot optimal caching files. The algorithms approach the optimal solution gradually with the user requesting. The main contributions of this work are summarized as follows:

- We practically model user preference and the change of user status with Zipf distribution and Markov chain respectively. Furthermore, the system is modelled as a Markov decision process (MDP), and the optimization objective, i.e., the content transmission energy consumption, is regarded as a criteria to supervise the learning process. The users change their status obeying the underlying probability rigorously and every status corresponds to a specific Zipf distribution instance.

- A RL method is introduced to determine the optimal caching policy on both user devices and SBS. Specifically, QL is applied on user device and DQN is applied on SBSs. A feedback mechanism based on the MDP is designed in this paper which generates training samples constantly. The RL algorithm is capable of adjusting the caching policy to address the optimal caching content. The preference of users will be predicted by the proposed RL algorithm after its convergence.
- The proposed scheme is extended to enable D2D communication between user devices. Once the D2D connection builds up, the cached popular content can be transmitted mutually between the users with lower energy consumption link. As a result, the systematic energy consumption can be further reduced. On the other hand, the D2D connection is constrained by the user location and the caching content.
- Numerical results validates the effectiveness of the proposed RL-based algorithm. More importantly, our findings have demonstrated that a significant energy saving can be achieved by our proposed RL-based algorithms, and this has confirmed the advantages of integrating D2D into cache-enabled IoT.

II. Preliminaries

In this section, the system model of D2D-assisted cache-enabled IoT is introduced first. Then, the energy cost formulations for different situations are provided. Finally, the mathematical formulation of the optimization problem is presented.

A. System Model

Consider multiple SBSs within the coverage of a macro base station (MBS), there exists a large number of users and devices in this area. Each user can cache M files and SBS can cache L files. There are F files in the core network. Due to the storage constraint of users and SBSs, $M \ll L \ll F$. Fig. 1 depicts the IoT scenario, users distribute randomly and they can take D2D to acquire files with relatively lower power, they can also store some popular and reusable files to reduce requests to the SBS and MBS. Similarly, users obtain files from SBS consumes much less energy than from MBS.

The requesting and satisfying procedure will be completed in one time slot t $(t = t_1, t_2, ...)$, where each slot is composed of three parts, i.e. content delivery, information exchange and content placement. In the content delivery stage which occupies most of the slots, users request files through D2D, SBS and MBS in turn as their demands. Then, in the information exchange stage, SBS collects all the requests submitted by the users and decides which files (L) to be cached in SBS. In the last stage, the selected files for the next slot will be placed in the storage of SBS, while users will cache M files. The last two stages will be carried out in off-peak time and the length of a time slot may be different depending on the network traffic situation.

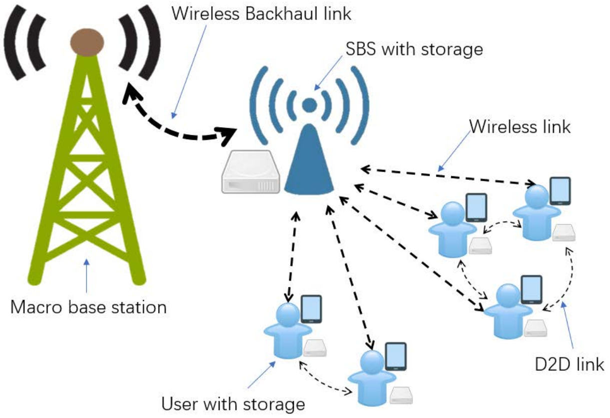


Fig. 1. A D2D-assisted cached-enabled IoT where nearer users employ D2D communication scheme.

While in the delivery stage of t, user u_i request file f_v, where $u_i \in \mathcal{U}$ and $f_v \in \mathcal{F}$. Specifically, $\mathcal{U} := \{u_1, u_2, ..., u_k\}$ represents the total users in the SBS coverage and $\mathcal{F} := \{f_1, f_2, ..., f_F\}$ denotes the total available files. Note $a(i,t) \in \mathcal{A}$ as the $1 \times F$ binary action vector where

$$a(i,t)[v] = \begin{cases} 1, & f_v \text{ is cached by } u_i \text{ in slot } t \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

and $\mathcal{A} := \left\{ a | a \in \{0,1\}^F \right\}$, therefore, $a\mathbf{1} = M$ for users. Similarly, $a_{SBS}(t) \in \mathcal{A}$ represents the caching action of SBS and $a\mathbf{1} = L$.

For user u_i, its behaviour can be described by the $1 \times F$ popularity vector $P_{t,i}$, the element $P_{t,i}[v] = P_{t,i,v}$ means the probability of the file f_v requested by u_i which can be observed at the end of content delivery stage as:

$$P_{t,i,v} := \frac{\text{user } u_i's \text{ requesting times for file } f_v \text{ in t}}{\text{user } u_i's \text{ total requesting times in t}} \quad (2)$$

Every user owns its unique popularity vector associated with the user's preference. Therefore, for user u_i, its status can be denoted as a $1 \times 2F$-vector $s(t)$ and $s(t) := [P_{t,i}, a_{i,t}]$, which will be used for user device's decision making. Furthermore, $\mathcal{S}$ denotes the state set for users. It should be noted that the action $a_{i,t}$ is carried out in slot t but it will influence the energy cost in slot $t + 1$. In general, our goal is to predict the optimal caching content for SBSs and user devices. Offloading the most popular content will reduce the energy consumption in next time slot, which will lead to a universal energy minimization.

B. Energy Consumption for Traffic Transmission

Consider path loss and small-scale fading first, channel gain h attenuates with distance d as d^e, where the path loss exponent e is usually assumed to be between 2 and 7 [30]. Furthermore, by taking small scale fading into account, the channel power gain h can be defined as $h = \theta_0 d^e |g|^2$,where $|g|^2$ is the small scale fading and $g \sim CN(0,1)$ is an independent and identically distributed circularly symmetric complex Gaussian vector with zero mean and covariance 1 [31]. The received power p_r can be written as $p_r = p_t h$. For user u_i, the requested content traffic with $F \times R_i > 0$ bits indicate its total demand in t. Based on these assumption, three available methods are listed as follows:

1) D2D Method: In the content delivery stage, users check if they have cached the requested files first, they use cached files of $R_{i,s}$ bits directly without any energy cost. Otherwise, they check if their available neighbours have cached the files, if the files have been cached by the neighbours, they will obtain the files by D2D communications. Due to the low transmit power constraint, the condition is very harsh for D2D that users have to be close enough, and hence we set the distance threshold as d_{max}. If $d_{u-u} \leq d_{max}$, the D2D link can be built up between the two users [32], if $d_{u-u} > d_{max}$, the D2D connection would not be built. According to Shannon's theory, the transmission data rate r of user u_i is

$$r = B\log_2(\mathit{1} + \frac{ph}{\sigma^2}), \tag{3}$$

where B is the transmission bandwidth, and σ^2 denotes additive white Gaussian noise. The cost time can be derived as the ratio of transmission traffic to transmission rate [33]:

$$t_{i,D} = \frac{R_{i,D}}{r_{i,D}}, \tag{4}$$

where $R_{i,D}$ bits denotes the traffic transmitted through D2D and p_D denotes the transmit power between D2D users. Therefore, the energy cost can be formulated as the product of cost time and transmit power [33], [34]:

$$E_{i,D} = t_{i,D} p_D = \frac{p_D R_{i,D}}{r_{i,D}}. \tag{5}$$

2) SBS Method: As same as D2D method, we can formulate data rate between user u_i and SBS with different distance d_{SBS-u} and transmit power p_{SBS}. Similarly, the energy cost can be written as:

$$E_{i,SBS} = \frac{p_{SBS} R_{i,SBS}}{r_{i,SBS}}, \tag{6}$$

where $R_{i,SBS}$ bits denotes the traffic that user u_i can not get from D2D but SBS.

3) MBS Method: For the case that the SBS cannot satisfy user u_i's demand, the rest request will be taken by MBS, the energy cost is

$$E_{i,MBS} = \frac{p_{MBS} R_{i,MBS}}{r_{i,MBS}}, \tag{7}$$

where p_{MBS} is the transmit power of MBS, and $R_{i,MBS}$ denotes the rest requested traffic.

4) Circuit Energy Consumption: Although there are traffic transmission in the information exchange stage and content placement stage of slot t, the two stages are too short to accumulate energy consumed. The content delivery stage may maintain for hours account for the network situation, but the last two stages just continue for seconds. Therefore, the energy cost of the last two stages will be ignored for simplicity. On the contrary, we cannot ignore the circuit energy consumption, due to the long term work time of circuit. The circuit energy cost can be written as [33], [34]:

$$E_{i,cir} = p_{cir} T, \tag{8}$$

where p_{cir} is the small circuit power and T is the time slot length in second.

C. Power Minimization Formulation

Based on the knowledge above, the power minimization problem of the D2D-assisted cache-enabled IoT can be mathematically formulated as:

$$\min_{u_i \in \mathcal{U}} \sum_{i=1}^{\mathcal{K}} (E_{i,D} + E_{i,SBS} + E_{i,MBS} + E_{i,cir}) \tag{9a}$$

$$s.t.\ R_{i,s} + R_{i,D} + R_{i,SBS} + R_{i,MBS} = FR_i, \forall i \in \mathcal{K}, \tag{9b}$$

$$\sum_{i=1}^{\mathcal{K}} (\frac{R_{i,D}}{r_{i,D}} + \frac{R_{i,SBS}}{r_{i,SBS}} + \frac{R_{i,MBS}}{r_{i,MBS}}) \leq T, \tag{9c}$$

$$r_{i,D} \geq 0,\ r_{i,SBS} \geq 0,\ r_{i,MBS} \geq 0, \tag{9d}$$

$$R_{i,s} \geq 0,\ R_{i,D} \geq 0,\ R_{i,SBS} \geq 0,\ R_{i,MBS} \geq 0. \tag{9e}$$

Among them, (9a) defines the objective function of the optimization problem; and (9b) is the total traffic amount constraint of one user, where $R_{i,s}, R_{i,D}, R_{i,SBS}$ and $R_{i,MBS}$ are bit-sized offloading amount of different edge nodes, and they vary with users' physical locations and caching selection strategies. Furthermore, the specific values of offloading amount are set to be non-negative in (9e). Similarly, the transmission rate is set to be non-negative in (9d). In (9c), the time constraint of K users is formulated, the total requests of users should be satisfied in every time slot.

We can observe that the objective function (9a) is rather complicated, and the constraints defined in (9b) and (9c) involve the caching policy and user locations, which are both discrete non-convex constraints. Thus, the problem (9a) is a non-convex optimization problem and difficult to obtain the optimal solution directly. In the following sections, we will develop RL-based schemes to jointly optimize caching policies in order to minimize universal energy cost.

III. Selection Decision and Reinforcement Learning

In this section, file popularity distribution is introduced first, and Markov chain that features user status follows. Then the objective function is detailed. The optimal condition of caching policy is drawn finally.

A. Zipf law distribution

We can make an reasonable assumption that the preference of user obeys Zipf law distribution, which is known as a famous model to measure the files popularity [35]. Furthermore, many realistic data set verifies the accuracy of Zipf law distribution [32]. There is an essential parameter γ in Zipf law, which differentiates the relative popularity of the files [35] as:

$$P_v = \frac{1}{v^\gamma \sum_{l=1}^{F} 1/l^\gamma}, \tag{10}$$

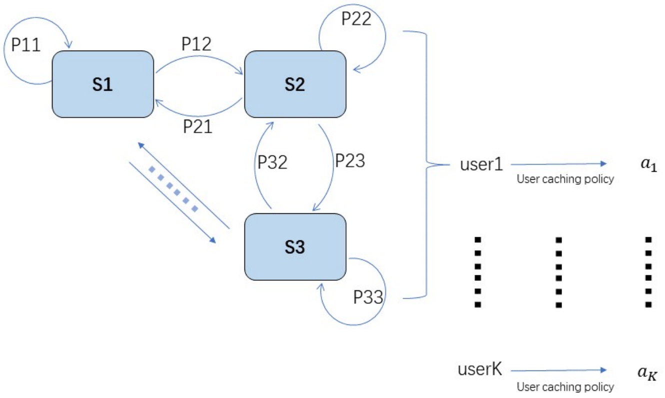


Fig. 2. The Markov chain illustration for user, the status changes with underlying probability.

which depicts the popularity of the v th most popular file. We can use (10) to replace (2) to simulate all the preference of files for users. Chen *et al.* had demonstrated that the best fitted distribution of one specific user is a Zipf distribution with parameter $\gamma = 1.05$ in [32], therefore, we assume that $\gamma \sim N(1, 0.5)$ by default throughput this work.

B. Markov Chain

As shown in Fig. 2, user's request behaviour is modelled by Markov chains, where each user changes status with their own probability. Each time slot, user occupied one status, and for every status, user have different request behaviour. In other words, users own different preference on files in different status. It's a practical assumption that user's request behaviour changes with relative long term but stay invariant in short term. Therefore, we can take user's preference invariant during a time slot [32] [36]. In this paper, user devices and SBS try different actions and learn the optimal policy from the surroundings. The Markov chain will generates independent sequential status constantly.

To characterize a Markov decision process, there are five fundamental elements: status set $\mathcal{S}$, action set $\mathcal{A}$, transition probability set $\mathcal{P}$, reward set $\mathcal{R}$ and discount factor η, among them, transmission probability $P_{12} \in \mathcal{P}$ denotes the conditional probability of state from S1 to S2, which is illustrated in Fig. 2. In addition, $\mathcal{A}$ denotes the caching action, $\mathcal{P}$ is unknown, and $\eta \in [0, 1]$ features the effect of historical data.

C. RL Formulation

For simplicity, we omit the lower corner t. In Fig. 2, assuming there are $|\mathcal{S}|$ states for one user, and the whole states for SBS can be written as a $1 \times 2F$ binary vector set $\mathcal{Q} := \{q|q = [c_1, c_2, ..., c_v, ..., c_F, a^\intercal], c_v = \sum_{i=1}^{K} p_{i,v} \ for \ v \in \mathcal{F}, a \in \mathcal{A}\}$. We can figure out the total energy cost for SBS as:

$$\begin{aligned}
E_{total}(t, a_{t-1}) &= \sum_{i=1}^{\mathcal{K}} (E_{i,cir} + E_{i,D} + E_{i,SBS} + E_{i,MBS}) \\
&= \sum_{i=1}^{\mathcal{K}} \Big\{ p_{cir} T + p_t \Big(R_i \sum_{e=1}^{\mathcal{F}} (\mathbf{1} - a_{i(t-1)}) \\
&\quad \mathbb{I}(\sum_{n=1}^{\mathcal{K}} a_{n(t-1)}[e] \geq 1) \mathbb{I}(d_{i-n} < d_{max}) \Big) \Big/ r_{i,D} \\
&\quad + p_{SBS} \Big(R_i \sum_{e=1}^{\mathcal{F}} (\mathbf{1} - a_{i(t-1)}) \mathbb{I}(\sum_{n=1}^{\mathcal{K}} a_{n(t-1)}[e] = 0, \\
&\quad d_{i-n} < d_{max}) \mathbb{I}(a_{SBS}[e] = 1) \Big) \Big/ r_{i,SBS} \\
&\quad + p_{MBS} \Big(R_i \sum_{e=1}^{\mathcal{F}} (\mathbf{1} - a_{i(t-1)}) \mathbb{I}(\sum_{n=1}^{\mathcal{K}} a_{n(t-1)}[e] = 0, \\
&\quad d_{i-n} < d_{max}) \mathbb{I}(a_{SBS}[e] = 0) \Big) \Big/ r_{i,MBS} \Big\}. \qquad (11)
\end{aligned}$$

Next, we formulate the cost function for users. Given the fact that the complexity of one user is relatively simple, we put forward a new criteria to measure the cost of each user. For the caching results, the more popular the cached file is, the less energy the user consumes. In other words, the cost is inversely proportional to the popularity of the cached file if the caching storage is limited. Moreover, the sum of different files' popularity is equal to 1, and hence we adopt the popularity to model the cost of user i as

$$cost_i = 1 - \sum P_i \Big[\underset{index}{\arg} \ a_{i,t-1}[index] = 1 \Big], \qquad (12)$$

which represents the general feature and clarify the cost of different actions. Therefore, we take this criteria for users' caching decision making.

Now, we can define our policy function $\pi = \mathcal{S} \to \mathcal{A}$ for user and $\pi = \mathcal{Q} \to \mathcal{A}$ for SBS. Thus we define a caching performance function named state-value function [35].

$$V_{total,\pi}(q(t)) := \lim_{T \to \infty} \mathbb{E} \Bigg[\sum_{\tau=t}^{T} \eta^{\tau - t} E_{total}(\tau, \pi(q[\tau])) \Bigg], \qquad (13)$$

$$V_{i,\pi}(s_t) := \lim_{T \to \infty} \mathbb{E} \Bigg[\sum_{\tau=t}^{T} \eta^{\tau - t} cost_i(\tau, s(\tau)) \Bigg], \forall i \in \mathcal{U}. \qquad (14)$$

It is an expected average reward under policy π over infinite time. According to (13), the value $V_\pi(s(t))$ depicts the influence of cost owing to current action and historical action. On the other hand, it also represents the uncertainties and imperfections. This discount factor avoids the large deviation due to erroneous data. In general, our purpose is to find out the optimal caching policy π^*_{user} for user and π^*_{SBS} for SBS. The energy cost of each user is minimum with the optimal caching policy on user devices and SBSs, therefore, the systematic energy cost keeps minimum if the optimal caching policy are taken. The optimal policies can be described as:

$$\pi^*_{user} = \arg \min_{\pi_{user} \in \Pi_{user}} V_{user,\pi}(s), \forall i \in \mathcal{U}, \forall s \in \mathcal{S}, \qquad (15)$$

$$\pi^*_{SBS} = \arg \min_{\pi_{SBS} \in \Pi_{SBS}} V_{total,\pi}(q), \forall q \in \mathcal{Q}. \qquad (16)$$

Equation (15) and (16) are both sequential decision making problems. We will present optimal solution in next section and introduce a QL method for solving problem (15) and a DQN method for problem (16).

IV. Optimal Solution and RL Methods

In this section, Bellman equations is first introduced, and then policy formulas are transformed to iterative forms which can be solved by RL, where the QL and DQN algorithms are presented in detail.

A. Bellman Equation

As a classic paradigm, dynamic programming deals with MDP well. Bellman equations formulate the main idea of dynamic programming [37]. Therefore, the recursive form of state-value function by using Bellman equation are:

$$\begin{aligned} V_{total,\pi}(q) &:= E_{total}(t, a_{t-1}) \\ &+ \eta \sum_{q_{t+1} \in \mathcal{Q}} P^{\pi(q)}_{q,q_{t+1}} V_{total,\pi}(q(t+1)), \forall q, q_{t+1} \in \mathcal{Q}, \end{aligned} \tag{17}$$

$$\begin{aligned} V_{i,\pi}(s) &:= cost_i \\ &+ \eta \sum_{s \in \mathcal{S}} P^{\pi(s)}_{s,s_{t+1}} V_{i,\pi}(s(t+1)), \forall s, s_{t+1} \in \mathcal{S}. \end{aligned} \tag{18}$$

The equation (17) consists of the observed cost $E_{total}(t-1, a_{t-1})$ and a discount of future state-value. $P^{\pi(s)}_{s,s_{t+1}}$ denotes the transition probability unknown in reality, and $\pi(s)$ is the action generated under a specific policy. Based on (17) and (18), the cost functions (11), (12) can be rewritten as:

$$E_{total}(t, a_{t-1}) = \sum_{q_{t+1} \in \mathcal{Q}} P^{\pi(q)}_{q,q_{t+1}} E_{t+1,a_t}, \tag{19}$$

$$cost_i = \sum_{s_{t+1} \in \mathcal{S}} P^{\pi(s)}_{s,s_{t+1}} cost_{i,t+1}. \tag{20}$$

With the above equations, we can obtain $V_{total,\pi}(q)$ and $V_{i,\pi}(s)$ with transition probability $P^{a}_{state,next\_state}$, and obtain the optimal policy π^* using policy iteration algorithm [37]. Here, we define state-action value function based on the underlying optimal policy, which is known as "Q-function":

$$Q_\pi(q, a_t) := E_{total}(t, a_{t-1}) + \eta \sum_{q_{t+1} \in \mathcal{Q}} P^{\pi(q)}_{q,q_{t+1}} V_{total,\pi}(q_{t+1}), \tag{21}$$

$$Q_\pi(s, a_t) := cost_i + \eta \sum_{s_{t+1} \in \mathcal{S}} P^{\pi(s)}_{s,s_{t+1}} V_{i,\pi}(s_{t+1}). \tag{22}$$

In order to achieve the ability to learn automatically, we design the updating steps as following [35]:

1) evaluating result: obtain $V_{i,\pi}(s)$ and $V_{total,\pi}(q)$ according to (17) and (18) based on the policy π for all status.

2) update policy: renew the policy with equation

$$\pi_{t+1}(state) := \arg\min_{\alpha} Q_{\pi_t}(state, \alpha). \tag{23}$$

Bellman equations formulate the optimal conditions for our problem but the transition probability is actually unknown in practice. Therefore, we can't compute (17) and (18) directly. In order to obtain an acceptable result, RL algorithms are taken into consideration. For user devices, since the amount of total request is usually limited in reality, a basic QL scheme with considerable low computation complexity and small model volume is suitable. On the other hand, since the situation is much complex for SBS, several deep RL algorithms are considered. The asynchronous advantage actor-critic (A3C) algorithm, which is proposed by Mnih $et\ al.$ in [38], has better convergence properties and is effective in high-dimensional or continuous action spaces. However, the A3C algorithm typically converges to a local optimum, which makes it inefficient for evaluating a policy. Another classic deep RL algorithm is proximal policy optimization (PPO) algorithm, which is proposed by Schulman $at\ al.$ in [39]. However, due to the parameters are updated along the direction of the policy gradient, there is also a limitation for PPO approach. Specifically, the parameter itself has its own spatial structure, and the direction of the strategy gradient does not take into account the spatial structure of the parameter itself, thus the update speed would be very slow. On the contrary, DQN is an off-policy algorithm with a relatively simple structure. More importantly, the application of experience reply accelerates the training process of DQN, and hence the training sample data can be utilized more efficiently. As a result, taking the complexity of the algorithm as well as the convergence speed into consideration, we employ DQN as our main algorithm to solve the caching file selection problem.

B. QL and DQN for Caching

QL is a classic RL algorithm to gradually approach the optimal selecting policy π^*, with evaluating the optimal state-action value function $Q^*(state, next\_action) := Q_{\pi^*}(state, next\_action), \forall state, next\_action$. Similar to the work in [35] and [37]. we can figure out that the optimal policy $\pi^*(s)$ satisfies

$$\pi^*(s) = \arg\min_{\alpha} Q^*(s, \alpha), \forall s \in \mathcal{S}. \tag{24}$$

Considering (14), we can combine Q-function and state-value function under π^* as

$$V^*(s) := V_{\pi^*}(s) = \min_{\alpha} Q^*(s, \alpha). \tag{25}$$

On the contrary, we can also get Q^* as

$$Q^*(s, a_t) = cost_i + \eta \sum_{s_{t+1} \in \mathcal{S}} P^{a}_{s,s_{t+1}} \min_{\alpha \in \mathcal{A}} Q^*(s_{t+1}, \alpha) \tag{26}$$

The agent in QL algorithm updates the estimated Q value as the real cost observed at the end of time slot. Given the last-slot state $s(t-1)$, action $a(t)$ and state $s(t)$, the cost can be described as $cost_i$. Meanwhile, the instantaneous error is

$$\begin{aligned} &\epsilon(s(t-1), a(t)) \\ &:= \frac{1}{2}\big(cost_i + \eta \min_{\alpha} Q(s(t), \alpha) - Q(s(t-1), a(t))\big)^2 \end{aligned} \tag{27}$$

according to the gradient descent algorithm, we can get the iteration equation as:

$$\begin{aligned} &Q(s(t-1), a(t)) = Q(s(t-1), a(t)) \\ &- \beta_t \frac{\partial}{\partial Q(s(t-1), a(t))} \epsilon(s(t-1), a(t)) \\ &= Q(s(t-1), a(t)) \\ &- \beta_t \frac{\partial \left(\frac{1}{2}\big(cost_i + \eta \min_{\alpha} Q(s(t), \alpha) - Q(s(t-1), a(t))\big)^2 \right)}{\partial Q(s(t-1), a(t))} \end{aligned} \tag{28}$$

Algorithm 1 User Caching via QL

```
Set $s_0$ randomly and initialize a $|\mathcal{S}||\mathcal{A}| \times |\mathcal{A}|$ table Q for every user and $Q_0(s,a) = 0, \forall s, a$
for t=1,2,..., do
    for $i \in \mathcal{U}$ do
        Choosing $a(t,i)$ by $\epsilon$-greedy algorithm
        $a(t,i) = \begin{cases} \arg\min_{a} Q_{t-1}(s_{t-1}, a_{t-1}) & \text{with } \epsilon \\ random\ a \in \mathcal{A} & \text{with } 1-\epsilon \end{cases}$
        $s$ is revealed at the end of slot as $[P_{t,i}, a_{t,i}]$
        Evaluate the cost value: $cost_i$
        Update Q value:
        $Q_{i,t}(s(t-1), a(t)) = (1-\beta_t) Q_{t-1}(s(t-1), a(t)) + \beta_t \big[cost_i + \eta \min_{\alpha} Q_{t-1}(s(t), \alpha)\big]$
    end for
end for
```

$$
\begin{aligned}
&= Q(s(t-1), a(t)) \\
&- \beta_t \big(cost_i + \eta \min_{\alpha} Q(s(t), \alpha) - Q(s(t-1), a(t))\big)(-1) \\
&= (1-\beta_t) Q(s(t-1), a(t)) + \beta_t \big[cost_i + \eta \min_{\alpha} Q(s(t), \alpha)\big].
\end{aligned}
$$

Based on the optimal conditions (24) – (28), the QL algorithm can be presented in **Algorithm 1**.

The Q value is updated using stochastic gradient descent algorithm [35], there are some necessary conditions to guarantee the result of QL approaching the optimality. In this work, we continuously renew the Q-table in order to satisfy these necessary conditions.

Due to the update mechanism that renew one value of the table each slot, the convergence speed is slow and the algorithm may trapped with the dimension disaster. Therefore, it is available only in simply scenario, and we use it on users only. Compared with QL, the neural-networks-based DQN is a better algorithm. The neural networks acts as the action-value function, and estimates the Q value directly with experience. The proposed DQN algorithm is presented in **Algorithm 2**.

Algorithm 2 SBS caching via DQN

```
Initialize reply memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$ and target action-value function $\hat{Q}$ with random weights $\theta^- = \theta$
for $episode = 1, 2, \ldots, E$ do
    Initialize sequence $s_1$ and preprocessed sequence $\phi_1 = \phi(s_1)$
    for t=1,2,..., do
        $a_{SBS} = \begin{cases} \arg\min_{\alpha} Q(\phi(s_t), \alpha; \theta) & \text{with } \epsilon \\ random\ a \in \mathcal{A} & \text{with } 1-\epsilon \end{cases}$
        Carry out $a_{SBS}$ in emulator and observe cost $E_{total}$
        Set $s_{t+1} = s_t, a_{SBS}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
        Store transition $(\phi_t, a_{SBS}, E_{total}, \phi_{t+1})$ in $D$
        Sample random minibatch of transitions $(\phi_j, a_{SBS}, E_{total}, \phi_{j+1})$ from $D$
        Set $y_j = \begin{cases} E_{total}, \text{episode end at step } j + \mathit{1} \\ E_{total} + \eta \min_{\alpha} \hat{Q}(\phi_{j+1}\alpha; \theta^-), \text{ otherwise} \end{cases}$
        Perform a gradient decent step on $(y_j - Q(\phi_j, a_{SBS}; \theta))^2$ with respect to the network parameter $\phi$
        Every $C$ steps reset $\hat{Q} = Q$
    end for
end for
```

The main idea of DQN is to build two neural networks where one stays steady and the other keeps evolution. In particular, the steady one update its parameter every C steps. The major advantage of DQN is that there exists no Q-table such that the performance of DQN is mainly rely on the accuracy of Q-value estimating via neural networks.

V. Simulation Results

The performance gain of the proposed RL-based joint scheme will be evaluated in this section. Three caching schemes are used as benchmark for comparison in the same simulation environment, which are listed in detail as follows:

- *Proposed*: The proposed joint scheme with QL on users and DQN on SBS, and D2D is applied between users' devices.
- *Optimal*: The popularities of all files is known a prior. The SBS chooses the most popular files for the whole coverage, and the user device chooses the most popular files for users in every slot. It's the theoretical optimal solution as well as the base line situation which measures the performance of other schemes.
- *Random Selection*: Both user devices and SBSs select the caching files randomly, and D2D communication works between users' devices. It takes all the caching action and D2D method except the RL algorithms compared with the proposed one. This control group can measure the effectiveness of the learning RL algorithms.
- *Without User Caching*: There is only DQN running on SBS, and there is no caching content on user devices. Therefore there is no D2D between users. With this control group, we can figure out the effectiveness of D2D and user local caching. Moreover, the effectiveness of DQN can be evaluated in a more standard environment in which there is no effect of user caching decisions.

Now, a two status Markov chain is defined for simplicity, i.e. every user change between two status independently with transition probability

$$
\mathbf{P}_{s,s_{t+1}} := \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}.
$$

Every user will change status according to the underlying transmission matrix independently. Then, the default parameters used in simulation are listed in TABLE I. To implement the algorithm, a well-known programming tool namely Tensorflow v1.11 is used on the Python 3.6.7 platform, nowadays, many works are programmed with Python and TensorFlow platform [40], [41]. The users are distributed in a square field with a side length of 120 meters, the SBS is on one corner of this square and the MBS is 100 meters far from the SBS. For DQN, the network is a 3 layers full connected neural network

TABLE I
Simulation Parameters

Name	Value
Learning rate β (DQN)	0.0001
Learning rate β (QL)	0.25
Discount rate η (DQN)	0.35
Discount rate η (QL)	0.8
Noise power σ^2	1e-9 W
Circuit power p_{cir}	1e-4 W
Power of D2D transmit p_t	0.05 W
Power of SBS transmit p_{SBS}	0.1 W
Power of MBS transmit p_{MBS}	40 W
D2D distance threshold d_{max}	30 m
Time slot length T	20 S
File size R_i	1e4 bit
Bandwidth B	2 MHz

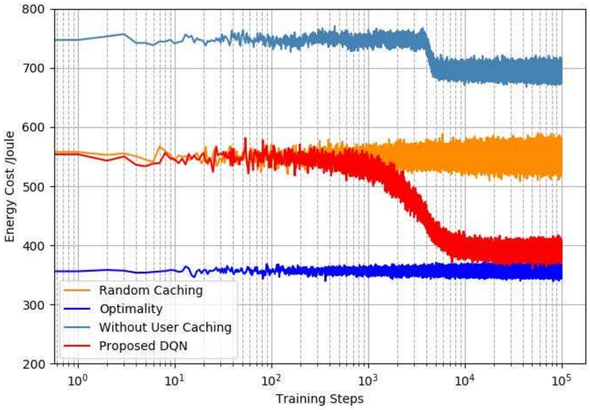


Fig. 3. Energy cost illustration with training steps for different schemes.

with random initialled weights and bias. The node numbers are 50, 135, 10 by order. The activation function is “ReLu” and the optimizer is “RMSPropOptimizer”. In order to train the DQN and QL algorithms on line, we implement the simulations by imitating the real environment. Particularly, the users request files along with their preference, and the status change is along with the transition probability matrices which is unknown for user devices and SBS. At the end of each slot, the requests of all users and the energy cost are exposed for SBS, and the DQN will make a caching decision for next slot based on the observed requests and energy cost. In order to create the “experience”, the observed requests, the caching action and the energy cost will be integrated as an experience sample to store in the memory storage. With the training process going on, the experience samples will be replaced by new samples gradually. For every epoch, the DQN selects a quantitative samples for training, and the energy cost corrects the Q value function along with the training, thus the estimation of the caching selection is getting more accurate. On the other hand, since the Q table in QL algorithm is not large, only $|\mathcal{S}||\mathcal{A}| \times |\mathcal{A}|$ values to update, the situation is much simpler for QL. It should be noted that these system parameters are merely chosen to demonstrate the energy saving performance in an example and can easily be modified to any other values depending on the specific scenario under consideration.

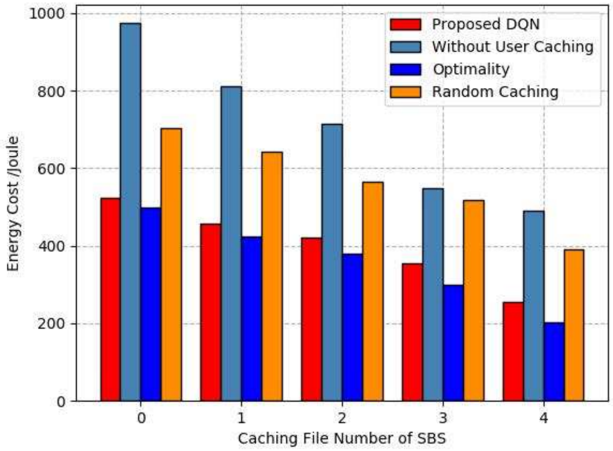


Fig. 4. Energy cost illustration with caching file number on SBS.

In the first set of simulation, the convergence of the proposed joint RL-based algorithms is studied over 30 realizations. The energy cost achieved by the proposed joint RL-based scheme is compared with that of the other three methods. As shown in Fig. 3, it can be observed that the energy cost of proposed solution converges about 5000 times to a stable value which is close to the value based on optimal solution. The result verifies the theoretical analysis where the proposed RL-based scheme is efficient compared with the optimal scheme as well as the random selection scheme. The optimal scheme chooses the most popular files every slot, and hence it reaches the theoretical edge. As a result, the cost of the optimal scheme remains the smallest throughout the simulation rounds. Furthermore, the stability of the optimal scheme is better than all the other schemes. Although there is no caching on users in without user caching scheme, there exists a decrease of around 5000 times, the whole characteristic parameters for one user are 200 (2×10×10), and hence the parameters have reached 2800 for the whole system. Therefore, the DQN network can observe all the users' feature after approximate 2800 times. Before that, the DQN network can not update the parameters effectively. In addition, due to the state transition behaviour of users, it needs more rounds to expose all the features. Therefore, a drop shows at around 5000 times in Fig. 3 for without user caching scheme, where the DQN algorithm has learnt the preference of total users in the region thoroughly and the prediction of SBS is accurate. For random selection scheme, the cost keeps relative stable statistically. However, due to the file selection mechanism, the random fluctuation can be visually observed in Fig. 3, which is larger than other three schemes. From Fig. 3, we can observe the impact of user's local caching. Specifically, even the random selection scheme has an obvious energy saving compared to the without user caching scheme.

Then we investigate the energy saving performance of proposed scheme with different file number of caching file on SBS. In this simulation, the parameters of constraints are similar to the previous, and the caching file number M on user device is fixed to 1 as default except the without user caching scheme. The number of caching files on SBS varies from 0 to 4. From Fig. 4, it is obvious that energy cost monotonically

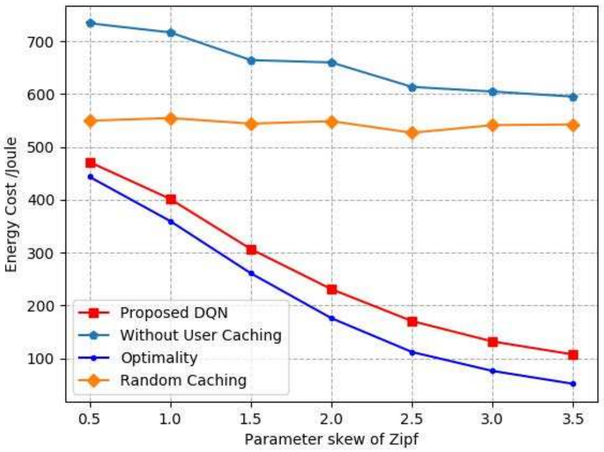


Fig. 5. Energy cost illustration with different user preference with γ.

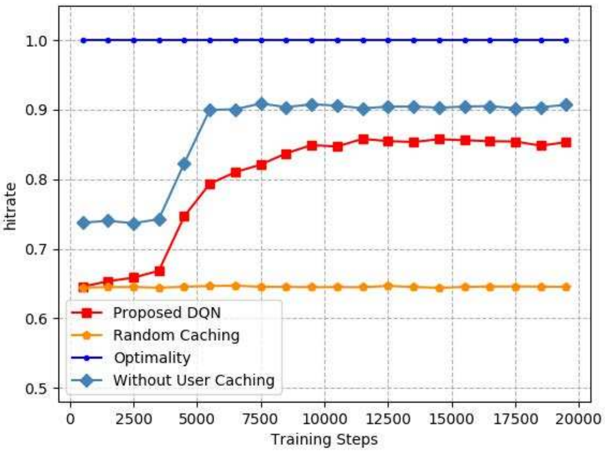


Fig. 6. Content hit rate of SBS with steps

decreases as caching file number increases. This is because the increasing caching files on SBS directly decreases the total requests to MBS, which leads to less energy consumption. On the other hand, for a fixed caching number of file on SBS, the optimal scheme keeps the best performance, and the proposed scheme has a similar performance to the optimal scheme. When there is no caching file on SBS, i.e. $L = 0$, the corresponding result shows the effect of QL and cache-enabled D2D. In particular, compare the proposed scheme with the random selection scheme, the gap is mainly depending on RL algorithms. The main difference between the proposed scheme and the random selection scheme depends on the caching files selection policies. For random selection scheme, the caching files are selected randomly, therefore, there is no prediction ability for both SBS and user devices. On the contrary, the proposed scheme applied RL algorithms on SBS and user devices, and hence they can learn the popularity of files and improves the quality of caching files selection. Besides, the prediction is getting accurate with additional training. Therefore, the gap between the proposed scheme and the random selection scheme is resulted by the RL algorithms. In addition, due to the DQN algorithm is able to choose popular files accurately, the without user caching scheme has the largest decrease amplitude.

In the next simulation, the energy saving performance of the proposed joint RL-based scheme under various user preference is evaluated and presented in Fig. 5. According to Zipf law distribution in (10), the skew parameter γ characterizes the user's preference uniquely. Specifically, it differentiates the preference on files, the larger γ is, the more the user prefers certain files. On the contrary, if $\gamma = 0$, the user have uniform preference on every file. As it can be seen in Fig. 5, the random selection scheme keeps nearly steady due to the randomness of the caching policy. The other three schemes decrease monotonically with the intensifying of user preference difference. For without user caching scheme, the amplitude attenuation is much smaller than the optimal and proposed schemes. This is because the change of individual user preferences have no obvious influence on the overall decision-making. Moreover, the proposed scheme traces the optimal solution rigorously. However, with the increase of γ, the cost of incorrect caching increases, the difference between expectation and every-slot optimal scheme gets bigger, therefore, the observed gap is more obvious. In addition, the gap between the proposed and without user caching scheme illustrates the impact of cache-enabled D2D. We can conclude that with the increase of γ, the performance of device caching and D2D is getting better. In addition, the increase of γ results in the excessive preference for users. Specifically, for a small amount of users, the request that SBS collected has no obvious emphasis on some files, and hence the DQN on the SBS can not get a significant energy saving. However, the increase of γ results in the excessive preference difference for users, which is benefit for QL algorithm to extract the vital preference. Therefore, we can observe the performance gap between the proposed scheme and the without user caching scheme.

Finally, we evaluate the percentage of requested frequency on selected cached content to the requested frequency on optimal cached content. The optimal scheme caches the most popular content every slot, therefore it keeps the optimal performance. It can be observed in Fig. 6 that the performance of random caching scheme stays stable statistically around 64.5 %. For the proposed scheme, the content hit rate increases more than other schemes, but it finally keeps stable around 85 %. Moreover, the performance of without user caching scheme is outperformed the proposed solution, which is caused by the disturbance of user devices caching content. Owing to the cached content on user devices, the submitted request to the SBS lacks of the cached part of user devices, and thus, the received popular distribution of proposed scheme is different from the one of without user caching scheme for SBS. With popular file-distribution, the prediction accuracy raises dramatically, which makes the content hitrate up to 90%.

VI. Conclusion

This paper has proposed a RL approach for D2D-assisted cache-enabled IoT, where the aim is to minimize the energy cost of systematic traffic transmission. To achieve this goal, we employed MEC for small cells to offload traffic from MBS. In addition, cache-enabled D2D communications has been introduced to small-cell users in order to further reduce the transmission energy cost. For the considered D2D-assisted cache-enabled IoT, MDP and Zipf distribution are employed to model the users requesting behaviors and the users' preference respectively. A novel RL-based algorithm has been proposed to reveal the popularity of files as well as users' preference, which are the key criteria for caching strategy. Specifically, we proposed to apply QL algorithm and DQN algorithm to users and SBS respectively in order to obtain an efficient caching policy. In addition, a feedback mechanism based on the MDP is developed in this paper which generates the training samples constantly. The proposed RL-based algorithm enables users' devices and SBS to prefetch the optimal files while learning, and hence reducing the energy cost significantly. Numerical results verified the effectiveness of the proposed RL-based algorithm. More importantly, our findings have demonstrated that a significant energy saving can be achieved by our proposed algorithm, and this has confirmed the advantages of integrating D2D communication into cache-enabled IoT. In addition, it has been shown in literature that the mobility of users could be studied in order to consider a practical scenario. Moreover, exploring the low complexity algorithm of solving the caching selection decision problem is also worthy of studying. Therefore, it will be a great value to investigate the low complexity solution considering the user mobility issue in the future.

References

[1] F. Boccardi, R. W. H. Jr, A. Lozano, T. L. Marzetta, and P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[2] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[3] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 756–764, IEEE, 2015.

[4] J. Song, S. Min, T. Q. S. Quek, X. Chao, and X. Wang, "Learning Based Content Caching and Sharing for Wireless Networks," *IEEE Transactions on Communications*, vol. 65, no. 10, pp. 4309–4324, 2017.

[5] K. Guo, C. Yang, and T. Liu, "Caching in Base Station with Recommendation via Q-Learning," in *Wireless Communications & Networking Conference*, pp. 1–6, IEEE, 2017.

[6] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing Dynamic Content in Caches with Small Population," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[7] L. Dong 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, 2015.

[8] S. Hakola, T. Chen, J. Lehtomaki, and T. Koskela, "Device-To-Device (D2D) Communication in Cellular Network - Performance Analysis of Optimum and Practical Communication Mode Selection," in *2010 IEEE wireless communication and networking conference*, pp. 1–6, IEEE, 2010.

[9] X. Liug, Z. Li, N. Zhao, W. Meng, G. Gui, Y. Chen, and F. Adachi, "Transceiver Design and Multi-hop D2D for UAV IoT Coverage in Disastersy," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1803–1815, 2019.

[10] G. Chen, J. Tang, and J. P. Coon, "Optimal Routing for Multihop Social-Based D2D Communications in the Internet of Things," *IEEE Internet of Things Journal*, vol. 5, pp. 1880–1889, June 2018.

[11] Y. Li, C. Zhong, M. C. Gursoy, and S. Velipasalar, "Learning-Based Delay-Aware Caching in Wireless D2D Caching Networks," *IEEE Access*, vol. 6, pp. 77250–77264, 2018.

[12] D. Wang, Y. Lan, T. Zhao, Z. Yin, and X. Wang, "On the Design of Computation Offloading in Cache-Aided D2D Multicast Networks," *IEEE Access*, vol. 6, pp. 63426–63441, 2018.

[13] R. Atat, L. Liu, N. Mastronarde, and Y. Yang, "Energy Harvesting-Based D2D-Assisted Machine-Type Communications," *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1289–1302, 2017.

[14] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8440–8450, 2018.

[15] X. Sun, G. Gui, Y. Li, R. P. Liu, and Y. An, "ResInNet: A Novel Deep Neural Network with Feature Re-use for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, pp. 679–691, Feb 2019.

[16] S. Li, J. Xu, M. Van Der Schaar, and W. Li, "A Transfer Learning Approach for Cache-Enabled Wireless Networks," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[17] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online Deep Reinforcement Learning for Computation Offloading in Blockchain-Empowered Mobile Edge Computing," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 8050–8062, Aug 2019.

[18] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, and X. Liang, "UAV Autonomous Target Search Based on Deep Reinforcement Learning in Complex Disaster Scene," *IEEE Access*, vol. 7, pp. 117227–117245, 2019.

[19] D. Zhao, Haitao Wang, Kun Shao, and Y. Zhu, "Deep reinforcement learning with experience replay based on SARSA," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–6, Dec 2016.

[20] S. Zhou, X. Liu, Y. Xu, and J. Guo, "A Deep Q-network (DQN) Based Path Planning Method for Mobile Robots," in *2018 IEEE International Conference on Information and Automation (ICIA)*, pp. 366–371, Aug 2018.

[21] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–5, Aug 2018.

[22] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, pp. 257–265, June 2018.

[23] S. Li, J. Xu, M. Van Der Schaar, and W. Li, "Popularity-Driven Content Caching," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[24] B. Chen, C. Yang, and Z. Xiong, "Optimal Caching and Scheduling for Cache-Enabled D2D Communications," *IEEE Communications Letters*, vol. 21, no. 5, pp. 1155–1158, 2017.

[25] X. Zhang, Y. Wang, R. Sun, and D. Wang, "Clustered device-to-device caching based on file preferences," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2016.

[26] G. Chen, J. P. Coon, A. Mondal, B. Allen, and J. A. Chambers, "Performance Analysis for Multihop Full-Duplex IoT Networks Subject to Poisson Distributed Interferers," *IEEE Internet of Things Journal*, vol. 6, pp. 3467–3479, April 2019.

[27] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.

[28] Y. He, C. Liang, F. R. Yu, and V. C. M. Leung, "Integrated Computing, Caching, and Communication for Trust-Based Social Networks: A Big Data DRL Approach," *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2018.

[29] F. Dong, T. Wang, and S. Wang, "Power Consumption Minimization in Cache-Enabled Mobile Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 6917–6925, July 2019.

[30] T. S. Rappaport, "Wireless Communications : Principles and Practice," 2002.

[31] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 1784–1797, March 2018.

[32] B. Chen and C. Yang, "Caching Policy for Cache-enabled D2D Communications by Learning User Preference," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6586–6601, 2018.

[33] X. Lin, H. Zhang, H. Ji, and V. C. M. Leung, "Joint computation and communication resource allocation in mobile-edge cloud computing networks," in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pp. 166–171, Sep. 2016.

[34] T. Sigwele, P. Pillai, and Y. Hu, “Saving Energy in Mobile Devices Using Mobile Device Cloudlet in Mobile Edge Computing for 5G,” in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 422–428, June 2017.

[35] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, “Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-time Popularities,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 180–190, 2018.

[36] D. Rafailidis and A. Nanopoulos, “Modeling Users Preference Dynamics and Side Information in Recommender Systems,” *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 46, no. 6, pp. 782–792, 2016.

[37] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction,” *IEEE Transactions on Neural Networks*, vol. 9, pp. 1054–1054, Sep. 1998.

[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, pp. 1928–1937, 2016.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[40] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, pp. 41–50, Feb 2018.

[41] X. Jin and H. Kim, “Deep Learning Detection in MIMO Decode-Forward Relay Channels,” *IEEE Access*, vol. 7, pp. 99481–99495, 2019.

Jie Tang (S10M13-SM18) received the B.Eng. degree in Information Engineering from the South China University of Technology, Guangzhou, China, in 2008, the M.Sc. degree (with Distinction) in Communication Systems and Signal Processing from the University of Bristol, UK, in 2009, and the Ph.D. degree from Loughborough University, Leicestershire, UK, in 2012. He is currently an associate professor at the School of Electronic and Information Engineering, South China University of Technology, China. He previously held Postdoctoral research positions at the School of Electrical and Electronic Engineering, University of Manchester, UK. His research interests include green communications, NOMA, 5G networks, SWIPT, heterogeneous networks, cognitive radio and D2D communications. He is currently serving as an Editor for IEEE Access, EURASIP Journal on Wireless Communications and Networking, Physical Communications and Ad Hoc & Sensor Wireless Networks. He also served as a track co-chair for IEEE Vehicular Technology Conference (VTC) Spring 2018. He is a co-recipient of the 2018 IEEE ICNC Best Paper Award.

Hengbin Tang received his B. Eng. degree in the school of Electronic and Information Engineering at the South China University of Technology, Guangzhou, China, in 2019. He is currently pursuing his M.Sc. at the School of Electronic and Information Engineering, South China University of Technology, China, under the supervision of Dr Jie Tang. His research interests include machine learning, mobile edge caching, mobile edge computing, multiple-input and multiple-output and 5G networks.

Xiuyin Zhang (S07-M10- SM12) received the B. S. degree in communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2001, the M.S. degree in electronic engineering from South China University of Technology, Guangzhou, China, in 2006, and the PhD degree in electronic engineering from City University of Hong Kong, Kowloon, Hong Kong, in 2009.

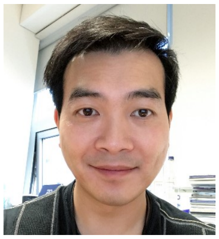
From 2001 to 2003, he was with ZTE Corporation, Shenzhen, China. He was a Research Assistant from July 2006 to June 2007 and a Research Fellow from September 2009 to February 2010 with the City University of Hong Kong. He is currently a full professor and vice dean with the School of Electronic and Information Engineering, South China University of Technology. He also serves as the deputy director of Guangdong Provincial Engineering Research Center of Antennas and RF techniques and the vice director of the Engineering Research Center for Short-Distance Wireless Communications and Network, Ministry of Education. He has authored or coauthored more than 100 internationally referred journal papers including 55 IEEE Transaction papers as well as around 60 conference papers. His research interests include microwave circuits and sub-systems, antennas and arrays, SWIPT.

Dr. Zhang is a Fellow of the Institution of Engineering and Technology. He has served as a Technical Program Committee (TPC) chair/ member and session organizer/chair for a number of conferences. He is an associate editor for the IEEE Access. He was a recipient of the National Science Foundation for Distinguished Young Scholars of China, the Young Scholar of the Chang-jiang Scholars Program of Chinese Ministry of Education, the Top-notch Young Professionals of National Program of China. He was also the recipient of the Scientific and Technological Award (First Honor) of Guangdong Province. He was the supervisor of several conference best paper award winners.

Kanapathipillai Cumanan (M’10-SM’19) received the BSc degree with first class honors in electrical and electronic engineering from the University of Peradeniya, Sri Lanka in 2006 and the PhD degree in signal processing for wireless communications from Loughborough University, Loughborough, UK, in 2009. He is currently a lecturer at the Department of Electronic Engineering, The University of York, UK. From March 2012 to November 2014, he was working as a research associate at School of Electrical and Electronic Engineering, Newcastle University, UK. Prior to this, he was with the School of Electronic, Electrical and System Engineering, Loughborough University, UK. In 2011, he was an academic visitor at Department of Electrical and Computer Engineering, National University of Singapore, Singapore. From January 2006 to August 2006, he was a teaching assistant with Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka. His research interests include non-orthogonal multiple access (NOMA), massive MIMO, physical layer security, cognitive radio networks, convex optimization techniques and resource allocation techniques.

Dr. Cumanan was the recipient of an overseas research student award scheme (ORSAS) from Cardiff University, Wales, UK, where he was a research student between September 2006 and July 2007.

Gaojie Chen (S'09-M'12-SM'18) received the B.Eng. and B.Ec. degrees in electrical information engineering and international economics and trade from Northwest University, China, in 2006, and the M.Sc. (Hons.) and Ph.D. degrees in electrical and electronic engineering from Loughborough University, Loughborough, U.K., in 2008 and 2012, respectively. From 2008 to 2009, he was a Software Engineering with DT mobile, Beijing, China, and from 2012 to 2013, he was a Research Associate with the School of Electronic, Electrical and Systems Engineering, Loughborough University. He was a Research Fellow with 5GIC, Faculty of Engineering and Physical Sciences, University of Surrey, U.K., from 2014 to 2015. Then he was a Research Associate with the Department of Engineering Science, University of Oxford, U.K., from 2015 to 2018. He is currently a Lecturer with the Department of Engineering, University of Leicester, U.K. He has served as an Editor for IET Electronics Letters (2018-present). His current research interests include information theory, wireless communications, cooperative communications, cognitive radio, secrecy communication, and random geometric networks.



Kai-Kit Wong (M'01-SM'08-F'16) received the BEng, the MPhil, and the PhD degrees, all in Electrical and Electronic Engineering, from the Hong Kong University of Science and Technology, Hong Kong, in 1996, 1998, and 2001, respectively. After graduation, he took up academic and research positions at the University of Hong Kong, Lucent Technologies, Bell-Labs, Holmdel, the Smart Antennas Research Group of Stanford University, and the University of Hull, UK. He is Chair in Wireless Communications at the Department of Electronic and Electrical Engineering, University College London, UK.

His current research centers around 5G and beyond mobile communications, including topics such as massive MIMO, full-duplex communications, millimetre-wave communications, edge caching and fog networking, physical layer security, wireless power transfer and mobile computing, V2X communications, and of course cognitive radios. There are also a few other unconventional research topics that he has set his heart on, including for example, fluid antenna communications systems, and team optimization. He is a co-recipient of the 2013 IEEE Signal Processing Letters Best Paper Award and the 2000 IEEE VTS Japan Chapter Award at the IEEE Vehicular Technology Conference in Japan in 2000, and a few other international best paper awards.

He is Fellow of IEEE and IET and is also on the editorial board of several international journals. He has served as Senior Editor for IEEE Communications Letters since 2012 and for IEEE Wireless Communications Letters since 2016. He is also Area Editor for IEEE Transactions on Wireless Communications. He had also previously served as Associate Editor for IEEE Signal Processing Letters from 2009 to 2012 and Editor for IEEE Transactions on Wireless Communications from 2005 to 2011. He was also Guest Editor for IEEE JSAC SI on virtual MIMO in 2013 and currently Guest Editor for IEEE JSAC SI on physical layer security for 5G.

Jonathon A. Chambers (S83M90SM98F11) received the Ph.D. and D.Sc. degrees in signal processing from the Imperial College of Science, Technology and Medicine (Imperial College London), London, U.K., in 1990 and 2014, respectively. From 1991 to 1994, he was a Research Scientist with the Schlumberger Cambridge Research Center, Cambridge, U.K. In 1994, he returned to Imperial College London as a Lecturer in signal processing and was promoted to a Reader (Associate Professor) in 1998. From 2001 to 2004, he was the Director of the Center for Digital Signal Processing and a Professor of signal processing with the Division of Engineering, Kings College London, where he is currently a Visiting Professor. From 2004 to 2007, he was a Cardiff Professorial Research Fellow with the School of Engineering, Cardiff University, Cardiff, U.K. From 2007 to 2014, he led the Advanced Signal Processing Group, School of Electronic, Electrical and Systems Engineering, Loughborough University, where he is also a Visiting Professor. In 2015, he joined the School of Electrical and Electronic Engineering, and he has been with the School of Engineering, Newcastle University, Newcastle upon Tyne, U.K., since 2017. Since 2017, he has also been a Professor in engineering and the Head of Department, University of Leicester, Leicester, U.K. He is also the International Honorary Dean and a Guest Professor with the Department of Automation, Harbin Engineering University, Harbin, China. He has advised almost 80 researchers through to Ph.D. graduation and has published over 500 conference proceedings and journal articles, many of which are in IEEE journals. His research interests include adaptive signal processing and machine learning and their application in communications, defense, and navigation systems.

Dr. Chambers is a fellow of the Royal Academy of Engineering, U.K., the Institution of Engineering and Technology, and the Institute of Mathematics and its Applications. He was a Technical Program Co-Chair of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, Czech Republic. He is serving on the Organizing Committees of ICASSP 2019, Brighton, U.K., and ICASSP 2022, Singapore. He has served on the IEEE Signal Processing Theory and Methods Technical Committee for six years, the IEEE Signal Processing Society Awards Board for three years, and the Jack Kilby Medal Committee for three year. He was an Associate Editor of the IEEE Transactions on Signal Processing for three terms over the periods 1997-1999, 2004-2007, and a Senior Area Editor from 2011 to 2015.