Incentive-driven Deep Reinforcement Learning for Content Caching and D2D Offloading

Huan Zhou, *Member, IEEE*, Tong Wu, *Student Member, IEEE*, Haijun Zhang, *Senior Member, IEEE*, and Jie Wu, *Fellow, IEEE*

***Abstract*—Offloading cellular traffic via Device-to-Device communication (or D2D offloading) has been proved to be an effective way to ease the traffic burden of cellular networks. However, mobile nodes may not be willing to take part in D2D offloading without proper financial incentives since the data offloading process will incur a lot of resource consumption. Therefore, it is imminent to exploit effective incentive mechanisms to motivate nodes to participate in D2D offloading. Furthermore, the design of the content caching strategy is also crucial to the performance of D2D offloading. In this paper, considering these issues, a novel Incentive-driven and Deep Q Network (DQN) based Method, named IDQNM is proposed, in which the reverse auction is employed as the incentive mechanism. Then, the incentive-driven D2D offloading and content caching process is modeled as Integer Non-Linear Programming (INLP), aiming to maximize the saving cost of the Content Service Provider (CSP). To solve the optimization problem, the content caching method based on a Deep Reinforcement Learning (DRL) algorithm, named DQN is proposed to get the approximate optimal solution, and a standard Vickrey-Clarke-Groves (VCG)-based payment rule is proposed to compensate for mobile nodes' cost. Extensive real trace-driven simulation results demonstrate that the proposed IDQNM greatly outperforms other baseline methods in terms of the CSP's saving cost and the offloading rate in different scenarios.**

***Index Terms*—D2D Offloading; Deep Reinforcement Learning; Reverse Auction; Content Caching; Real Mobility Trace.**

I. Introduction

THE explosive growth of smart devices and wireless service applications has not only brought great convenience to the society, but also brought tremendous mobile traffic to the mobile networks. Due to the large demand for a variety of content, the Content Service Provider (CSP) is put under great pressure to satisfy nodes' quality or experience requirements of services towards 5G cellular networks [1], [2], [3]. According to the recent report of Cisco, the global requested mobile traffic will reach 77.5 exabytes per month in 2022 [4]. Therefore, it is urgent for the CSP to be able to provide a quick and promising method to relieve the traffic load of cellular networks.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872221, and National Science Foundation (NSF) under Grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, and CNS 1564128. (Corresponding Author: H. Zhou)

H. Zhou, and T. Wu are with the College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China. E-mail: zhouhuan117@gmail.com, wutong.asd@gmail.com.

H. Zhang is with Institute of Artificial Intelligence, Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China. E-mail: haijunzhang@ieee.org.

J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia PA19122, USA. E-mail: jiewu@temple.edu.

Mobile data offloading is regarded as an effective way to relieve the traffic burden of cellular networks, which applies complementary network technologies to deliver mobile traffic that was originally planned to be transmitted via cellular networks [5], [6], [7]. Mobile data offloading can be implemented in many ways such as small base stations, Wi-Fi networks, heterogeneous networks, or Device-to-Device (D2D) communications. In recent years, data offloading via small base stations, Wi-Fi networks, and heterogeneous networks have evolved as mature technologies. However, they all rely on infrastructures, which have some disadvantages such as high maintenance cost, expensive installation cost, and limited coverage [8].

Another effective way to offload cellular traffic, is to deliver contents via D2D communications, also called D2D offloading [9], [10]. In D2D communications, nodes can employ their mobile devices with wireless interfaces for intermittent communication when they are within the mutual communications range [11], [12]. Different from the traditional content delivering method via cellular networks, D2D offloading first distributes the content to a small set of nodes, then these nodes can further help deliver the content to others who request the content via opportunistic D2D transmissions. Most mobile services provided by the CSP, such as multimedia newspapers, weather forecasts or advertisements, do not have strict real-time requirements and need to be delivered to a large number of nodes. Because of these non-real-time applications, the CSP only needs to transmit content to a small set of selected nodes, called caching nodes in this paper, under this way the cellular traffic and the operation cost of the CSP can be reduced.

Recent studies [13] have demonstrated that D2D offloading can effectively relieve the traffic burden of cellular networks. However, mobile nodes may not be willing to take part in D2D offloading without proper financial incentives (such as payment or reward) since the data offloading process will incur energy and transmission cost [14]. As a result, it is necessary to exploit effective incentive mechanisms to motivate nodes to participate in D2D offloading. Meanwhile, the design of the content caching strategy is also crucial to the performance of D2D offloading. First, nodes in D2D communications meet opportunistically; it is difficult to predict nodes' movement and mutual contact. Second, the amount of data transferred between mobile nodes is related to the contact times and the duration of contacts, thus one content may not be transmitted completely at one time. Third, although the cache capacity of mobile nodes has increased greatly, it is still limited compared to the CSP. Therefore, from the perspective of the CSP, it

remains open to jointly consider the content caching strategy and incentive mechanism to improve the performance of D2D offloading.

Based on the above analysis, the following issues are considered in this paper: (i) How to design efficient content caching strategy? (ii) How to stimulate mobile nodes to take part in D2D offloading and what is the corresponding payment of the CSP to each mobile node? (iii) How to maximize the saving cost of the CSP while satisfying some specific constraints? We answer these issues by proposing a novel Incentive-driven and Deep Q Network (DQN) based Method, named IDQNM. In IDQNM, the reverse auction is employed as the incentive mechanism, where the CSP acts as the auctioneer and mobile nodes act as the bidders. Then, the incentive-driven D2D offloading and content caching process is modeled as Integer Non-Linear Programming (INLP), aiming to maximize the saving cost of the CSP. To solve the optimization problem, the content caching method based on a Deep Reinforcement Learning (DRL) algorithm, named DQN is proposed to get the approximate optimal solution by exploiting Deep Neural Network (DNN). Furthermore, a standard Vickrey-Clarke-Groves (VCG)-based payment rule is proposed to compensate for mobile nodes' cost in D2D offloading.

The contributions of this paper are summarized as follows:

1) A dynamic and time-varying D2D offloading system is explored with consideration of the uncertain and dynamic content requests, mobility as well as the limited cache capacity of nodes.
2) An Incentive-driven and DQN-based Method, named IDQNM is proposed to stimulate nodes to participate in D2D offloading, in which the reverse auction is employed as the incentive mechanism, and a DQN-based method is proposed to solve the content caching optimization problem.
3) To compensate for mobile nodes' cost in D2D offloading, an innovative VCG-based payment rule is proposed, which guarantees the individual rationality and truthfulness properties of the proposed IDQNM.
4) The real trace-driven simulation results demonstrate that our proposed IDQNM greatly outperforms other baselines in terms of the CSP's saving cost and the offloading rate in different scenarios.

The rest of this paper is organized as follows. Section II reviews the related work, and Section III introduces the system model, including the network architecture, opportunistic D2D transmissions model and reverse auction model. In Section IV, the problem is formulated as INLP with the objective to maximize the saving cost of the CSP. Section V introduces the DQN-based content caching method, and the payment rule based on the standard VCG scheme. In Section VI, the individual rationality and truthfulness of the proposed IDQNM are proved. Section VII introduces the performance evaluation. Finally, Section VIII concludes this paper.

II. Related Work

Some studies have exploited data offloading through D2D communications from different perspectives. Pan et al. in [15] jointly considered the social characteristics and physical transmission, and proposed an iterative algorithm to maximize the offloaded traffic via D2D communications. In [16], the authors formulated the data offloading through D2D communications as a link prediction problem, and proposed a framework based on the link prediction which can reconstruct the observed network more realistic. From the energy perspective, Yang et al. in [17] formulated the cost-aware energy efficient data offloading problem as a discrete time optimal control problem. Due to the curse of dimensionality, an approximation based method was proposed to solve the problem. In [18], Yu et al. considered the social relationship in multi-access edge computing and proposed a Monte-Carlo based efficient task assignment method, named $TA-MCTS$, to minimize the energy consumption. Zhang et al. in [19] jointly considered the interference between mobile nodes, caching state, link scheduling and routing, and proposed an online algorithm based on the Lyapunov drift-plus-penalty theory to minimize the energy consumption through D2D communications. Zhao et al. in [20] proposed a social-aware three-phase method to improve the data offloading efficiency. Recently, some studies have applied reinforcement learning technology to improve the performance of data offloading. In [21], the authors investigated the system model, and proposed a multi-agent reinforcement learning-based cooperative content caching method to improve nodes' quality of experience, in which the preference of nodes are unknown and only the historical demands for the content can be observed. The authors in [22] designed a novel edge caching framework, and proposed an DDPG-based mechanism to reduce the system cost and the content delivery latency. In [23], a lightweight deep-learning technique was proposed to select the best channel for mobile nodes in D2D communications, which is different from the traditional methods that specify one channel on a specific band at a moment. Wang et al. in [24] proposed a heterogeneous collaborative edge caching framework based on D2D communications, which jointly optimizes the node selection and cache replacement. To solve the optimization problem, they proposed an attention-weighted federated DQN-based method to control the decision process. In [25], the authors considered the edge caching problem in hierarchical wireless networks and proposed a distributed content replacement strategy based on the Q-learning algorithm, which can be used in the large-scale real trace. However, the above studies assume that nodes are cooperative in providing data offloading services, and they do not consider the situation that nodes are selfish or rational.

Recently, some incentive mechanisms based on economic theory have been proposed to stimulate nodes to participate in data offloading, such as game theory [26], [27], contract theory [28] and auction theory [29] - [36]. Shah et al. in [26] modeled the interactions in the market as a Stackelberg game with three-stage, and discussed two games of different objective functions of the Mobile Network Operator (MNO). In [27], the authors modeled the interaction between mobile nodes as an incomplete information bargaining game, and proposed a distributed incentive mechanism to reduce the total amount of payment when multi-hop offloading is considered. Under the premise of guaranteeing the service quality, Chen

et al. in [28] designed a contract-based incentive mechanism to maximize the operator's expected profit by using D2D multicast communications. In [29], a novel mobile network data transaction system based on basic and networked auction models is proposed to lead highly efficient data allocation among nodes. In [30], Paris et al. designed a combinatorial reverse auction mechanism to select the cheapest Wi-Fi APs and offload the maximum amount of traffic from the MNO. In [31], Song et al. proposed a reverse auction-based incentive mechanism with a cost constraint in content distribution via D2D communications. In [32], a truthful double auction method was proposed for resources trading in multi-cell multi-channel networks to achieve good performance from the perspective of economy, and make both the buyer/seller obtain satisfactory benefit. In [33], the authors investigated the caching placement methods based on the multi-winner auction approach to reduce the content caching redundancy. Du et al. in [34] proposed a second-priced auction-based spectrum sharing and traffic offloading mechanism for the hybrid satellite-terrestrial networks, in which the truthful bid is the dominant strategy and the winners only need to pay the auctioneer the second highest price. The authors in [35] proposed a robust optimization algorithm based on the multi-item auctions when the MNO has incomplete information, which can guarantee the individual rationality, incentive ability, and budget feasibility. Du et al. [36] designed a double auction-based video caching mechanism to elicit the insufficient or hidden information, and maximize the social welfare in heterogeneous ultra-dense networks. However, the above existing studies about auction-based incentive mechanism in D2D offloading mainly formulate the auction from the perspective of users, where some users with contents act as sellers, and other requested users act as buyers.

Compared to the above previous studies, this paper jointly considers the content caching strategy and incentive mechanism to improve the performance of D2D offloading the perspective of the CSP. Furthermore, an Incentive-driven and DQN-based method, named IDQNM is proposed to stimulate nodes to participate in D2D offloading and maximize the saving cost of the CSP. In the design of IDQNM, the reverse auction is employed as the incentive mechanism, and a DQN-based method is used to select caching nodes.

III. System Model

This section introduces the system model related to our proposed method in detail. We consider the scenario with a CSP, a Base Station (BS), and some mobile nodes. These mobile nodes with limited cache capacity are within the coverage of the BS, and some contents need to be transferred from the CSP to the requested nodes within the deadline. The BS can receive the contents from the CSP through the backhaul wired link, then the BS can deliver the contents to mobile nodes through the cellular links. Meanwhile, the mobile nodes can deliver contents to each other via D2D communications.

TABLE I
Notations and Symbols

Notation	Explanation
$\mathcal{L}$	The set of time slot
T	The duration of each time slot
$\mathcal{F}$	The set of contents
D_f	The size of content f
$\mathcal{T}_f(l)$	The tolerant delay of content f at time slot l
$\mathcal{T}_f^0$	The maximum tolerant delay of content f
$\mathcal{N}$	The set of mobile nodes
$Z_i(l)$	Node i's cache capacity at time slot l
Z_i^0	Node i's maximum cache capacity
μ_{ij}	The transmission rate of node i to node j
$q_{jf}(l)$	The size of content f requested by node j at time slot l
$T_{ijf}^{trans}(l)$	The transmission time that node j's request for content f can be satisfied by node i at time slot l
$c_{if}(l)$	The size of content f that node i has cached at time slot l
e_{if}	Node i's interest to content f
$x_{if}(l)$	Binary variable indicates if node i is selected as a caching node for content f
λ_{ij}	The contact rate between nodes i and j
ϖ_{ij}	The number of contacts between nodes i and j
$P_{ijf}(\mathcal{T}_f(l), q_{jf}(l))$	The offloading probability that node j who requests content f can be served by node i within the tolerant delay $\mathcal{T}_f(l)$
$\tau_1(l)$	The total amount of traffic that should be transmitted by the CSP at time slot l
$\tau_2(l)$	The amount of traffic transmitted via D2D communications at time slot l
$\tau_3(l)$	The extra cellular traffic transmitted to caching nodes who are not interested in the contents at time slot l
α	The unit traffic cost via cellular links
v_i	True value of the unit cost in D2D communications of node i
b_i	The expected unit traffic price of caching node i
$B_{if}(l)$	Node i's expected compensation for offloading content f at time slot l
$R_{if}(l)$	The offloading potential of node i for content f at time slot l
$\delta_{if}(l)$	The cost that the CSP can save by offloading the content f through node i at time slot l
$U_{\mathcal{H}(l)}$	The CSP's saving cost after selecting caching nodes at time slot l
$\mathcal{H}$	The set of caching nodes selected by the CSP
M_i	The marginal contribution of caching node i
p_i	The real payment of caching node i
$\mathcal{P}$	The set of caching nodes' payments

A. The Content Model

The proposed system runs over an infinite time period, which is divided into L time slots, denoted as $\mathcal{L} = \{1, 2, ..., L\}$, and the duration of each time slot is T. We assume that the CSP has F contents, denoted as $\mathcal{F} = \{1, ..., F\}$, and the popularity of each content is different and follows the Zipf distribution. Moreover, the popularity of each content equals the preference of each node. Each content has limited data size, denoted as $\mathcal{D} = \{D_1, D_2, ..., D_F\}$. Different contents have different tolerant delays denoted as $\mathcal{T}_{\mathcal{F}}(l) = \{\mathcal{T}_1(l), \mathcal{T}_2(l), ..., \mathcal{T}_F(l)\}$, where $\mathcal{T}_f(l)$ represents the tolerant delay of content $f \in \{1, 2, ...F\}$ at time slot l. Mobile nodes can get the requested content via D2D communications before the deadline, otherwise it will be directly transmitted by the BS. $\mathcal{T}_f^0(f \in \{1, ...F\})$ denotes the tolerant delay of content f in the initial time. Therefore, $\mathcal{T}_f(l)$

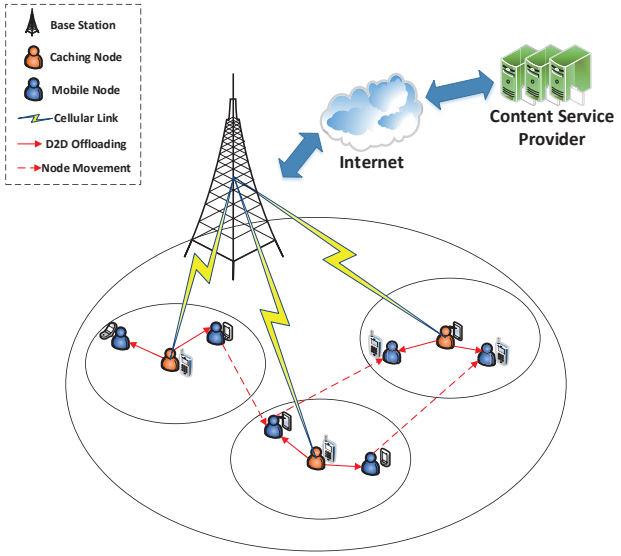


Fig. 1. Opportunistic D2D offloading scenario.

can be updated as follows:

$$\mathcal{T}_f(l) = \begin{cases} \mathcal{T}_f^0 - (l-1)T, & \mathcal{T}_f^0 - (l-1)T > 0 \\ 0, & \mathcal{T}_f^0 - (l-1)T \le 0. \end{cases} \tag{1}$$

It is worth noticing that when $\mathcal{T}_f(l) > 0$, the node's requests will be satisfied through D2D communications, otherwise the rest of the requests will be transmitted by the BS through the cellular links.

B. Mobile Node Model

N mobile nodes exist in the network, denoted as $\mathcal{N} = \{1, ..., N\}$. All mobile nodes are within the service coverage of the BS, and each mobile node has the following properties:

- **The cache capacity** $Z_i(l)$: $\mathcal{Z}(l) = \{Z_1(l), Z_2(l), ..., Z_N(l)\}$ is used to denote the available cache size of the nodes at time slot l, and $Z_i^0 (i \in \{1, 2, ..., N\})$ represents node i's cache capacity in the initial time.
- **Data transmission rate** μ_{ij}: Similar to [33], we use orthogonal model to allocate non-overlapping orthogonal radio resources for D2D transmissions, in which the bandwidth of each node is divided into equal sub-bands. When a caching node transmits contents to different nodes at the same time, each node is assigned an equal sub-band and there is no interference between different sub-bands. Then, the data transmission rate between nodes i and j can be given as:

$$\mu_{ij} = \frac{W_i}{n_i} \log\left(1 + \frac{h_{ij}(l) P_i^{Trans}}{\omega(l) + \sigma^2}\right), \tag{2}$$

where W_i is node i's bandwidth, n_i is the number of D2D communications pairs of node i, P_i^{Trans} denotes the transmission power of node i when delivering content via D2D communications, $\omega(l)$ denotes the co-channel interference levels from adjacent cells in time slot l, $h_{ij}(l)$ denotes the channel gains between nodes i and j in time slot l, and σ^2 denotes the additive white Gaussian noise power.

- **The size of the requested content** $q_{jf}(l)$: Let the continuous variable $q_{jf}(l)$ represent the size of content f requested by node j at time slot l. Then, the transmission time that node j's request for content f can be satisfied by node i at time slot l can be calculated as:

$$T_{ijf}^{trans}(l) = \frac{q_{jf}(l)}{\mu_{ij}}. \tag{3}$$

- **The size of the cached content** $c_{if}(l)$: Let $\mathbb{C}(l)$ denote the set of cache state of mobile nodes for each content, and the element of $\mathbb{C}(l)$ be represented by $c_{if}(l), i \in \mathcal{N}, f \in \mathcal{F}$, which denotes the size of content f that node i has cached at time slot l. It can be updated as follows:

$$c_{if}(l) = \begin{cases} e_{if}\left(D_f - q_{if}(l)\right), & x_{if}(l) = 0 \\ D_f, & x_{if}(l) = 1, \end{cases} \tag{4}$$

where $x_{if}(l)$ indicates whether node i is selected to cache content f at time slot l, and e_{if} indicates whether node i is interested in content f. If node i is interested in content f, $e_{if} = 1$, otherwise $e_{if} = 0$. When node i is selected to cache content f, the BS will deliver the content to node i via cellular links immediately. Then, the available cache capacity of node i at time slot l can be updated as follows:

$$Z_i(l) = Z_i^0 - \sum_{f=1}^{F} c_{if}(l). \tag{5}$$

C. Opportunistic D2D Transmissions Model

In D2D communications, node's movement and mutual contact are difficult to predict; it is really hard to estimate the delivery probability of content f even along a particular path [37]. Thus, in order to estimate the delivery probability, a probabilistic framework is proposed according to the contact pattern. We assume that a pair of nodes can contact each other multiple times within the time constraint, so the amount of data transferred between them is related to the number of contacts and the duration of the contacts. The contact duration of each node pair is modeled as the Pareto distribution, which is based on the statistics of each node pair [37]. It is worth noting that since different node pairs' distribution of contact duration and frequency are heterogeneous, node pairs have specific parameters for their distributions.

Similar to [37], $P\left(\mathbb{T}_{ij} \le \mathcal{T}_f(l)\right)$ is used to denote the probability that nodes i and j contact ϖ_{ij} times within the tolerant delay, where $\mathbb{T}_{ij} \sim \Gamma\left(\varpi_{ij}, \lambda_{ij}\right)$ is a random variable that represents the time required for ϖ_{ij} opportunistic contacts, and λ_{ij} is the contact rate of node pair i and j. Then, $P\left(\mathcal{G}_{ij} \ge q_{jf}(l)\right)$ is used to denote the probability that node j's request of content f can be delivered with ϖ_{ij} opportunistic contacts, where $\mathcal{G}_{ij}$ is the total amount of traffic delivered through ϖ_{ij} opportunistic contacts.

Let $\varpi_{ijf}^{max}(l)$ represent the maximum number of contacts that node j's request of content f at time slot l can be satisfied

by node i, then, the probability that node j's request of content f can be satisfied by node i within $\mathcal{T}_f(l)$ is calculated as:

$$P_{ijf}\left(\mathcal{T}_f(l), q_{jf}(l)\right) = \sum_{k=1}^{\varpi_{ijf}^{max}(l)} \widehat{P}_{k-1} \cdot P\left(\mathcal{G}_{ij}^{k} \geq q_{jf}(l)\right) \cdot P(\mathbb{T}_{ij}^{k} \leq \mathcal{T}_f(l) - T_{ijf}^{trans}(l)), \tag{6}$$

where $T_{ijf}^{trans}(l)$ denotes the transmission time that node j's request can be satisfied by node i at time slot l which is given in Eq. (3), and $\widehat{P}_{k-1}$ is the probability that $q_{jf}(l)$ cannot be completely transmitted in the first $k-1$ times, which can be formulated as follows:

$$\widehat{P}_{k-1} = \begin{cases} \prod_{h=1}^{k-1} P\left(\mathbb{T}_{ij}^{h} \leq \mathcal{T}_f(l) - T_{ijf}^{trans}(l)\right) \\ \cdot P\left(\mathcal{G}_{ij}^{k} < q_{jf}(l)\right) & k > 1 \\ 1 & k = 1. \end{cases} \tag{7}$$

The details of Eq. (6) can be found in the APPENDIX.

D. The CSP Model

The CSP transmits contents to mobile nodes via BS through cellular networks, and let α denote the unit cost of traffic through cellular links. $\tau_1(l)$ denotes the total cellular traffic that should be transmitted by the CSP before selecting caching nodes at time slot l. $\tau_2(l)$ denotes the amount of traffic transmitted via D2D communications after selecting caching nodes at time slot l. However, not all nodes are interested in content f; if a caching node has no interest in content f, the CSP needs to pay an extra cost through cellular links denoted as $\tau_3(l)$. Then, if the payment to caching nodes is not considered, the saving cost of the CSP after selecting caching nodes can be expressed as:

$$\begin{aligned} C(\tau_1(l), \tau_2(l), \tau_3(l)) =& C(\tau_2(l), \tau_3(l)) \\ =& \tau_1(l)\alpha - (\tau_1(l) - \tau_2(l) + \tau_3(l))\alpha \\ =& (\tau_2(l) - \tau_3(l))\alpha. \end{aligned} \tag{8}$$

E. The Mobile Nodes' Bidding Model

Mobile nodes will not be willing to provide data offloading services without any reward. We assume that each node i ($i \in \mathcal{N}$) has the following properties:

- **Node i's expected price in opportunistic D2D transmissions** b_i: b_i is node i's expected reward by providing data offloading services through D2D communications.
- **The true value of the unit cost in opportunistic D2D transmissions** v_i: v_i is the real cost consumed by node i for providing data offloading services, it should be noted that v_i is a private information of node i, which cannot be obtained by anyone else even the CSP. Moreover, each node has an individual rationality property, which ensures that each node can get a non-negative reward, so, it should be also noted that the true value v_i may not equal the node's bidding value b_i.

$\delta_{if}(l)$ is used to denote the saving cost of selecting node i to cache content f within the tolerant delay, which can be expressed as:

$$\delta_{if}(l) = \alpha\left(R_{if}(l) - D_f\left(1 - e_{if}\right)\right), \tag{9}$$

where $R_{if}(l)$ represents node i's offloading potential for content f at time slot l, which means the amount of data size of content f can be offloaded by node i. $R_{if}(l)$ can be given as:

$$R_{if}(l) = \sum_{j \in \mathcal{N} \setminus i} P_{ijf}\left(\mathcal{T}_f(l), q_{jf}(l)\right) q_{jf}(l). \tag{10}$$

$B_{if}(l)$ is used to denote the expected reward that node i wants to obtain in the process of offloading content f, which can be calculated as:

$$B_{if}(l) = R_{if}(l)\, b_i, \tag{11}$$

then $\delta_{if}(l) - B_{if}(l)$ denotes the actual saving cost after selecting node i to cache content f at time slot l.

F. Reverse Auction Model

This paper employs the reverse auction to motivate mobile nodes to take part in D2D offloading. Specifically, the CSP acts as an auctioneer that needs to employ nodes with cache capacity. At the beginning of each time slot, the CSP collects nodes' bids and the cache states. These cache states include the nodes' current cache states of content f ($f \in \mathcal{F}$) and nodes' available cache at time slot l. Then, according to the historical contact records and bids of nodes, the CSP will select proper caching nodes for each content, and caching nodes will obtain the corresponding rewards based on the size of data they delivered. The process of the reverse auction can be summarized as follows:

- The mobile nodes in the coverage of the BS submit their expected prices b_i to the CSP.
- At the beginning of each time slot, each mobile node reports the expected amount of content f ($f \in \mathcal{F}$) and the remaining available cache to the CSP. Then, based on the received information, the CSP selects some nodes as caching nodes for each content.
- When all requests are satisfied, the actual payment for each caching nodes will be calculated.

IV. Problem Formulation

In this section, the incentive-driven D2D offloading and content caching process is modeled as an optimization problem, aiming to maximize the saving cost of the CSP. At each time slot, the CSP needs to select caching nodes for each content f ($f \in \mathcal{F}$) according to the states. Let $x_{if}(l)$ denote whether node i is selected to cache content f at time slot l. If node i is selected to cache content f, $x_{if}(l) = 1$, otherwise $x_{if}(l) = 0$. Once node i is selected to cache content f, the BS will deliver the content to node i via cellular links immediately. Let the set $\mathcal{X}(l)$ contain all of the selecting variables at time slot l as $\mathcal{X}(l) = \{x_{if}(l) | i \in \mathcal{N}, f \in \mathcal{F}\}$, and let the set $\mathcal{Q}(l)$ contain all of the content requesting variables as $\mathcal{Q}(l) = \{q_{if}(l) | i \in \mathcal{N}, f \in \mathcal{F}\}$. Then, considering the payments

to the selected caching nodes, the CSP's expected saving cost at time slot l is formulated as:

$$\begin{aligned} & U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) \\ =& C\left(\tau_2(l), \tau_3(l)\right) - \sum_{i \in \mathcal{N}} \sum_{f \in \mathcal{F}} x_{if}(l) B_{if}(l) \\ =& C\left(\sum_{i \in \mathcal{N}} \sum_{f \in \mathcal{F}} (1 - x_{if}(l)) q_{if}(l),\right. \\ & \left. \sum_{i \in \mathcal{N}} \sum_{f \in \mathcal{F}} x_{if}(l)\left(1 - e_{if}\right)\left(D_f - c_{if}(l-1)\right)\right) \\ & - \sum_{i \in \mathcal{N}} \sum_{f \in \mathcal{F}} x_{if}(l) B_{if}(l), \end{aligned} \tag{12}$$

where $C(\cdot)$ denotes the CSP's expected saving cost without considering the payments to the caching nodes, which is shown in Eq. (8). According to the definition of $\tau_2(l)$, we should consider the sum amount of traffic that is requested by nodes except the caching nodes at time slot l, so we can set $x_{if}(l) = 0$ and use $\sum_{i \in N} \sum_{f \in \mathcal{F}} \left(1 - x_{if}(l)\right) q_{if}(l)$ to calculate $\tau_2(l)$; according to the definition of $\tau_3(l)$, we should consider the sum amount of traffic that are sent from the CSP to the caching nodes not interested in the content at time slot l, so we can set $x_{if}(l) = 1$, and $e_{if} = 0$. If node i has cached part of content f denoted as $c_{if}(l-1)$ in the past time slot, then the CSP only needs to transmit the left $D_f - c_{if}(l-1)$ to node i, so $\sum_{i \in N} \sum_{f \in \mathcal{F}} x_{if}(l)\left(1 - e_{if}\right)\left(D_f - c_{if}(l-1)\right)$ can be used to calculate $\tau_3(l)$. $\mathcal{H}(l)$ is the set of caching nodes at time slot l, and $B_{if}(l)$ means the expected reward that node i wants to obtain in the process of offloading content f which is shown in Eq. (11).

From the perspective of the CSP, it aims to maximize its saving cost, thus the optimization objective is formulated as:

$$\max \quad \sum_{l=1}^{L} U_{\mathcal{H}(l)}\left(\mathcal{X}(l), \mathcal{Q}(l)\right) \tag{13}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}} x_{if}(l)\left(D_f - c_{if}(l)\right) \leq \sum_{i \in \mathcal{N}} q_{if}(l), \forall f \in \mathcal{F}, \tag{14}$$

$$\sum_{f \in \mathcal{F}} c_{if}(l) \leq Z_i(l), \quad \forall i \in \mathcal{N}, \tag{15}$$

$$\sum_{f \in \mathcal{F}} e_{if} D_f \leq Z_i^0, \quad \forall i \in \mathcal{N}, \tag{16}$$

$$Z_i^0 - \sum_{f \in \mathcal{F}} e_{if} D_f \geq \sum_{f \in \mathcal{F}} (1 - e_{if}) x_{if}(l) D_f, \ \forall i \in \mathcal{N}, \tag{17}$$

$$b_i \leq \alpha, \quad \forall i \in \mathcal{N}, \tag{18}$$

$$x_{if}(l) \in \{0,1\}, \ e_{if} \in \{0,1\}, \quad \forall i \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{19}$$

$$q_{if}(l) = \left[0, D_f\right], \quad \forall i \in \mathcal{N}, \forall f \in \mathcal{F}, \tag{20}$$

$$\mathcal{T}_f(l) \geq 0, \quad \forall f \in \mathcal{F}, \tag{21}$$

where constraint (14) guarantees that at time slot l, the total traffic of content f ($f \in \mathcal{F}$) delivered by the CSP is not larger than the total amount of requested traffic; constraint (15) guarantees that the total amount of contents cached in node i cannot exceed its available capacity; constraint (16) guarantees the rationality of each node's request, the amount of requested content cannot exceed its own available cache capacity; constraint (17) guarantees that the CSP will consider node i's actual available cache capacity when selecting node i to cache content f; constraint (18) indicates that the caching nodes' expected rewards cannot be larger than the unit traffic cost of the CSP; constraint (19) guarantees the integer nature of binary variables; constraint (20) guarantees the rationality of requests; and constraint (21) guarantees that the tolerant delay of each content cannot be negative.

To solve the problem in Eq. (13), we should find the optimal content caching decision vector $\mathcal{X}(l) = \{x_{if}(l) | i \in \mathcal{N}, f \in \mathcal{F}\}$ at each time slot, where $x_{if}(l)$ is a binary variable. However, as the number of contents or nodes increases, the complexity of the problem increases exponentially. Thus, it can be found that the objective function is an INLP problem and belongs to NP-hard. Since it is hard to solve this problem by using traditional optimization methods, a DRL-based method is proposed in this paper.

V. DQN-based Content Caching Method

In this section, the content caching method based on a DRL algorithm, named DQN is introduced to resolve the problem above, and then the CSP's payment determination to the caching nodes is introduced.

A. State, Action and Reward Definition

The D2D offloading process is modeled as an MDP, and the definitions of each critical elements of MDP are given as follows:

- **System States**: The system states which reflect the environment consist of the request state, the cache state, and the available cache capacity. Let $q_{if}(l), i \in \mathcal{N}, f \in \mathcal{F}$ denote the request state of node i at time slot l, which can be observed to determine the total amount of requested contents. The cache capacity constraint also needs to be considered, while $c_{if}(l), i \in \mathcal{N}, f \in \mathcal{F}$ is used to denote the cache state of each node for each content, and $Z_i(l), i \in \mathcal{N}$ is used to denote the size of the available cache capacity of each node. Then, the state vector at time slot l is defined as:
$$S(l) = \{q_{if}(l), c_{if}(l), Z_i(l)\}, \quad i \in \mathcal{N}, f \in \mathcal{F}. \tag{22}$$
We use the set $\mathcal{S}$ to denote the finite state space, and we can get $S(l) \in \mathcal{S}, l \in \mathcal{L}$.
- **Action Space**: The CSP needs to select caching nodes for each content at time slot l. $x_{if}(l), i \in \mathcal{N}, f \in \mathcal{F}$ is used to denote the decision variables. Then, the action vector can be described as:
$$A(l) = \{x_{if}(l)\}, \quad i \in \mathcal{N}, f \in \mathcal{F}. \tag{23}$$
Let the set $\mathcal{A}$ represent the finite action space, from which we can get $A(l) \in \mathcal{A}, l \in \mathcal{L}$.
- **Reward**: The reward can be regarded as the feedback of an action taken by the agent. The saving cost function is used to denote the reward of the system, which is described as:
$$r\left(A(l), S(l)\right) = U_{\mathcal{H}(l)}\left(\mathcal{X}(l), \mathcal{Q}(l)\right). \tag{24}$$

B. Markov Decision Process

The MDP is the standard method of Sequential Decision Making (SDM) [38]. In our proposed model, the CSP adaptively learns and makes decisions by interacting with the environment at each time slot. For example, at time slot l the agent observes the current state $S(l)$ ($S(l) \in \mathcal{S}$), and decides to take an action $A(l)$ ($A(l) \in \mathcal{A}$) based on policy π, then the state will transfer into the next state $S'(l)$, and the agent will receive an immediate reward $r(A(l), S(l))$ after selecting action $A(l)$. Then, the probability that the current state $S(l)$ will transfer to the next state $S'(l)$ after selecting action $A(l)$ is defined as:

$$Pr\left(S'(l) \mid S(l), A(l)\right). \tag{25}$$

The state-value function is usually used to evaluate the long-term influence of a certain strategy at the current state. Considering that the influence of the reward will be smaller as it is far away from the current state, the cumulative discount expected rewards are used to represent the value of the current state, which can be formulated as:

$$V^{\pi}(S(l)) = E_{\pi}\left[\left(\sum_{l=0}^{L} \varphi^{l} r\left(A(l), S(l)\right)\right) \mid S(0) = S(l)\right], \tag{26}$$

where E represents the expectation, $\varphi \in (0, 1)$ represents the discounting factor which indicates the importance of the predicted expected reward, and $S(0)$ is the initial state.

Since MDP has the properties of the Markov Model, the state at the next time slot is only determined by the current state. Thus, according to the Bellman Equation, V-value is transformed as:

$$V^{\pi}(S(l)) = r(A(l), S(l)) + \varphi \sum_{S'(l) \in \mathbb{S}} Pr(S'(l) | A(l), S(l)) V^{\pi}(S'(l)). \tag{27}$$

It can be seen that our goal is to find an optimal strategy $\pi^*(S(l))$ to maximize the cumulative discounted reward, which can be formulated as:

$$\begin{aligned} &V^{\pi*}(S(l)) \\ &= \max_{\pi} \left[r(A(l), S(l)) + \varphi \sum_{S'(l) \in \delta} \Pr\left(S'(l) \mid A(l), S(l)\right) V^{\pi*}(S'(l)) \right] \\ &\text{s.t.} \quad (14)\ (15)\ (16)\ (17)\ (18)\ (19)\ (20)\ (21) \end{aligned} \tag{28}$$

Then, we can get the optimal action for state $S(l)$ under the policy $\pi^*(S(l))$:

$$A(l)^* = \underset{A(l)}{\operatorname{argmax}}\ V^{\pi}(A(l), S(l)). \tag{29}$$

C. DQN-based Solution

In the formulated MDP problem (28), it is hard to obtain the optimal policy $\pi^*(S(l))$ due to the large state space and action space. Therefore, a reinforcement learning algorithm, named Q-learning is first used to solve the formulated MDP problem. In fact, each strategy $\pi(S(l))$ corresponds to a series of actions. We disassemble a strategy into multiple actions and get the value function through a certain action, denoted as $Q(A(l), S(l))$, which is stored in a Q-table. In this paper, $Q(A(l), S(l))$ is used to estimate the optimal state value function $V^{\pi*}(S(l))$ under the state $S(l)$, and the relationship between them can be obtained as:

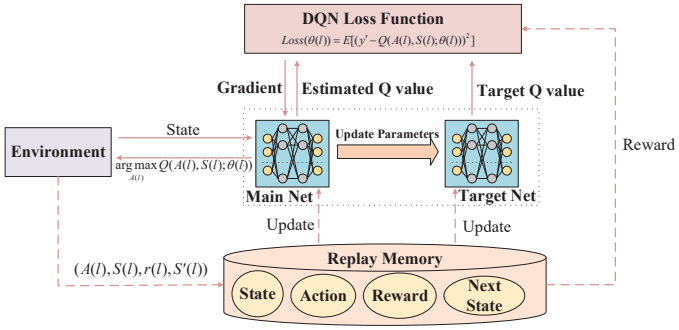


Fig. 2. DQN architecture for solving the proposed MDP problem.

$$\begin{aligned} &V^{\pi*}(S(l)) = \max_{A(l)} Q^{\pi}(A(l), S(l)) \\ &\text{s.t.} \quad (14)\ (15)\ (16)\ (17)\ (18)\ (19)\ (20)\ (21). \end{aligned} \tag{30}$$

Then, the expected cumulative reward from taking action $A(l)$ at state $S(l)$ is formulated as:

$$\begin{aligned} &Q^{\pi}(A(l), S(l)) \\ &= r(A(l), S(l)) + \varphi \sum_{S'(l)} p\left(S'(l) \mid S(l), A(l)\right) V^{\pi^*}(S'(l)). \end{aligned} \tag{31}$$

Combined with Eq. (30), Eq. (31) is updated as:

$$\begin{aligned} &Q^{\pi}(A(l), S(l)) \\ &= r(A(l), S(l)) + \varphi \sum_{S'(l)} p\left(S'(l) \mid S(l), A(l)\right) \max_{A'(l)} Q^{\pi}(S'(l), A'(l)). \end{aligned} \tag{32}$$

According to Eq. (32), the maximum state-action function $Q^{\pi*}(A(l), S(l))$ and the optimal selecting actions can be derived from the iteration of value and action. The update process of $Q^{\pi}(A(l), S(l))$ can be expressed as:

$$\begin{aligned} &Q^{\pi}(A(l), S(l)) \\ &= Q(A(l), S(l)) + \varepsilon[r(A(l), S(l)) + \varphi \max_{A'(l)} Q(A'(l), S'(l)) \\ &\quad - Q(A(l), S(l))] \end{aligned} \tag{33}$$

where ε is the learning rate and the update process of $Q(A(l), S(l))$ named the Q-learning process. The traditional Q-learning process uses a Q-table to store the state-action combinations and the related Q-values. However, the state of our proposed model is continuous, thus a finite Q-table cannot store the infinite amount of state-action values. In order to compensate for Q-learning's limitation, the Q-learning method is incorporated with deep learning technology named as the Deep Q-Network (DQN). Different from the Q-learning method, DQN-based method uses a DNN as a nonlinear approximator, which can capture complex interactions between various states and actions. Then, the optimal Q value can be approximated by the parameter θ of the DNN. The Q value in DQN is expressed as:

$$Q(A(l), S(l)) \approx Q(A(l), S(l); \theta), \tag{34}$$

where θ denotes the weight of the main neural network. Then, we can get the optimal action for caching node selection in state $S(l)$ with the maximum $Q(A(l), S(l); \theta)$, which is expressed as:

$$A(l)^* = \underset{A(l)}{\operatorname{argmax}}\ Q\left(A(l), S(l); \theta\right). \tag{35}$$

In order to ensure the approximation ability, $Q\left(A(l), S(l); \theta\right)$ should be trained via the value of the target neural network $r(A(l), S(l)) + \varepsilon \underset{A'(l)}{\max} Q\left(A'(l), S'(l)\right)$, then the estimated Q-value is obtained as:

$$y' = r(A(l), S(l)) + \varepsilon \max_{A'(l)} Q\left(A'(l), S'(l); \bar{\theta}(l)\right). \tag{36}$$

The goal of DQN is to obtain the minimum difference between the estimated value and the target value, thus the loss function is defined as:

$$Loss\left(\theta(l)\right) = \mathrm{E}\left[\left(y' - Q\left(A(l), S(l); \theta(l)\right)\right)^2\right], \tag{37}$$

where $\bar{\theta}(l)$ in Eq. (36) is the parameter from the previous time slot $l-1$. In order to update $\theta(l)$, we differentiate $Loss(\theta(l))$ with respect to the weight parameter $\theta(l)$, and derive the gradient as:

$$\nabla_{\theta(l)} Loss\left(\theta(l)\right) = \frac{\partial Loss(\theta(l))}{\partial \theta(l)}. \tag{38}$$

Then, $\theta(l)$ can be updated according to the gradient descent as:

$$\theta(l) \leftarrow \theta(l) - \eta \nabla_{\theta(l)} Loss\left(\theta(l)\right), \tag{39}$$

where η denotes the coefficient of updating step size.

Moreover, DQN introduces an experience replay mechanism to remove the correlations in the subsequent training samples and improve learning efficiency. A tuple $(A(l), S(l), r(l), S'(l))$ is used to denote the learned experience at time slot l, which is stored in the replay buffer to train the DNN's parameters at each time slot l. Throughout the training process, a batch of stored experiences as samples are randomly selected by the experience replay mechanism in DQN. During the training process, in order to balance exploration and exploitation, and also avoid local optimum, an $\epsilon-$greedy policy is used to select an action in DQN. The agent will explore better selection strategies by randomly selecting an action with probability ϵ, otherwise the action with the highest estimated Q-value is selected. Fig. 2 shows the architecture for solving the proposed MDP problem in DQN, in which two neural networks are considered, denoted as the Main Net and the Target Net. The estimated Q-value is calculated by using the Main Net, and the target Q-value is calculated by using the Target Net. In order to replace the cache efficiently, $M_{if}(l)$ is used to denote the marginal contribution of node i to content f, which is defined as:

$$M_{if}(l) = U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)), \tag{40}$$

where $U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l))$ denotes the optimal solution that node i does not participant in the offloading of content f. If caching node i does not have enough storage to cache the content, the caching node will automatically select the proper content to replace it based on the marginal contribution.

Algorithm 1 DQN-based Content Caching Method.

Initialize the replay memory
Initialize the Main Net with random weight θ
Initialize the Target Net with weight $\bar{\theta} = \theta$
for each episode **do**
  Calculate the delivery probability of each node pair;
  Obtain the initial observation O^0 and pre-process O_0 as the initial state $S(0)$;
  for each step of episode **do**
    Choose a random probability $\ni$
    if $\ni \leq \epsilon$ **then**
      randomly select an action $A(l)$
    else
      Select action $A(l) = \arg\max_{A(l)} Q\left(A(l), S(l); \theta(l)\right)$
    end if
    for $f \in \mathcal{F}$ **do**
      for $i \in \mathcal{N}$ **do**
        if $M_{if}(l) > 0$ **and** $x_{if}(l) = 1$ **and** $c_{if}(l-1) < D_f$ **and** $Z_i(l) < (D_f - c_{if}(l-1))$ **then**
          $g \leftarrow \arg\min_{k \in \mathcal{K}}(M_{ik}(l))$ **and** $M_{ik}(l) < M_{if}(l)$ **and** $c_{ik}(l-1) + Z_i(l) \geq D_f - c_{if}(l-1)$;
          Replace content g with content f, $c_{if}(l) = D_f$;
        end if
      end for
    end for
    Execute action $A(l)$, calculate the system reward $r(A(l), S(l))$ and receive the next observation $O'(l)$
    Pre-process $O'(l)$ to be the next state $S'(l)$;
    Store the experience $(A(l), S(l), r(l), S'(l))$ into the replay memory
    Randomly select a minibatch of samples $(A(j), S(j), r(j), S'(j))$ from the replay memory
    Calculate the target Q-value from the Target Net, $y'_j = r(A(j), S(j)) + \varepsilon \max_{A'(j)} Q\left(A'(j), S'(j); \bar{\theta}(j)\right)$;
    Perform the gradient descent step on $Loss(\theta(j))$ with respect to $\theta(j)$;
    Update the Main Net parameter $\theta(j)$ and the tolerant delay of each content;
    Update the Target Net parameter $\bar{\theta}(j)$ with $\theta(j)$ every $\tilde{M}$ steps;
    Calculate the delivery probability of each node pair for the next time slot.
  end for
end for

The more specific detail of the proposed DQN-based content caching method can be found in Algorithm 1.

D. Payment Determination

The CSP needs to determine the payment to compensate for the selected caching nodes' cost. The caching nodes' expected prices have already known to the CSP before selecting them, however, due to the nature of selfishness and rationality, each node wants to get a higher reward that is not equal to the real value they submitted. Therefore, it is necessary to formulate a

Algorithm 2 Payment Determination.

Require: $\mathcal{N}$, $\mathcal{F}$, b_i, $\mathcal{X}$, $\mathcal{Q}$;
Ensure: $\mathcal{P}$;
Initialization: $\mathcal{P} \leftarrow \emptyset$;
for $i \in \mathcal{H}$ **do**
  for $f \in \mathcal{F}$ **do**
    $p_{if} = 0$;
  end for
end for
for $f \in \mathcal{F}$ **do**
  for $i \in \mathcal{H}_f$ **do**
    Calculate $U^{-i}_{H_f}(\mathcal{X}, \mathcal{Q})$ according to Eq. (41);
    Let $x_{if} \equiv 0$;
    Update $\mathcal{H}$ according to the trained DQN-based content caching method;
    Calculate p_{if} according to Eq. (42);
    $\mathcal{P} \leftarrow \mathcal{P} \cup p_{if}$;
  end for
end for
return $\mathcal{P}$

uniform rule to ensure the rationality of payment. In this part, a VCG-based payment rule is introduced to motivate nodes to participate in D2D offloading and guarantee the nature of individual rationality and truthfulness.

In the standard VCG scheme, each bidder submits a quotation without knowing the bids of others. As introduced before, $\delta_{if}(l) - B_{if}(l)$ denotes the actual saving cost of the CSP after selecting node i as a caching node for content f. Let $U^{-i}_{\mathcal{H}_f}(\mathcal{X}, \mathcal{Q})$ denote the optimal solution without considering the contribution of node i for content f, which can be given as:

$$U^{-i}_{\mathcal{H}_f}(\mathcal{X}, \mathcal{Q}) = \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) - x_{if}(l)(\delta_{if}(l) - B_{if}(l))). \tag{41}$$

Moreover, let $U_{\mathcal{H}|(x_{if}\equiv 0)}(\mathcal{X}, \mathcal{Q})$ denote the optimal solution that node i does not participant in the offloading of content f, then we can get the actual payment to caching node i for offloading content f as:

$$p_{if} = \sum_{l=1}^{L} \left(\delta_{if}(l) x_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l))\right) + U^{-i}_{\mathcal{H}_f}(\mathcal{X}, \mathcal{Q}). \tag{42}$$

Let $\phi_{if} = v_i \sum_{l=1}^{L} x_{if}(l) R_{if}(l)$ represent the sum of the real cost consumed by caching node i for content f in D2D offloading. Then, we can get the utility of caching node $i \in \mathcal{H}$ for content f as:

$$u_{if} = p_{if} - \phi_{if}. \tag{43}$$

The details of the proposed payment determination are shown in Algorithm 2.

E. Complexity Analysis

Similar to [39], the computational complexity of algorithm 1 is related to the number of mobile nodes and the number of contents, as well as the training times and updating parameters. We use K and L to denote the total training episodes and the total number of time slots, respectively. The computational complexity of parameter updating of Main Net is the same as that of Target Net, so we use G and $\tilde{G}$ to denote the computational complexity of parameter updating and gradient decent, respectively. In IDQNM, the parameter of Target Net will be updated every $\tilde{M}$ steps, where $\tilde{M}$ is a constant we have set in advance. Then, the computational complexity of the proposed IDQNM is $O(K \times L \times (F \times N + \tilde{G} + G + G/\tilde{M}))$. Meanwhile, since the action space and state space of the input layer of our system are proportional to the number of mobile nodes in the network, when the number of nodes increases, the complexity of the IDQNM will also increase.

For algorithm 2, since the content caching method in algorithm 1 has been trained, the computational complexity of the Payment Determination is $O(H \times F)$.

VI. Theoretic Analysis

In this section, we prove that the proposed payment determination satisfies two important properties: individual rationality and truthfulness. The property of individual rationality guarantees that each selected caching node can get a non-negative compensation, which is the main motivation for nodes to take part in D2D offloading. The truthfulness property guarantees that the caching nodes cannot get a higher compensation from untruthful bids.

Theorem 1. *(Individual Rationality). The payment rule in Eq.* (42) *satisfies the individual rationality property, e.g.,* $\forall i \in \mathcal{H}_f, f \in \mathcal{F}, u_{if} \geq 0$.

Proof. According to the payment rule in Eq. (42), we obtain:

$$\begin{aligned} p_{if} &= \sum_{l=1}^{L} \left(\delta_{if}(l) x_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l))\right) + U^{-i}_{\mathcal{H}_f}(\mathcal{X}, \mathcal{Q}). \\ &= \sum_{l=1}^{L} \left(\delta_{if}(l) x_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l))\right) \\ &\quad + \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) - (\delta_{if}(l) - B_{if}(l)) x_{if}(l)). \\ &= \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \\ &\quad + B_{if}(l) x_{if}(l)). \end{aligned}$$

We assume that if caching node $i \in \mathcal{H}_f$ bids truthfully, i.e., $B_{if} = \phi_{if}$, we get:

$$\begin{aligned} u_{if} &= p_{if} - \phi_{if} \\ &= \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l))) - \sum_{l=1}^{L} (U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l))) \\ &\geq 0. \end{aligned}$$

Therefore, through the analysis above, the proposed method satisfies the property of individual rationality. □

Theorem 2. *(Truthfulness). Eq.* (42) *is the payment rule which guarantees the truthfulness property. It can be proved that it*

is a weakly dominant strategy for each selected caching node to set the bid $b_i = v_i$.

Proof. If node i is selected as the caching node for content f and submits the bid $b_i^{'}$ untruthfully, i.e., $b_i^{'} \neq v_i$. Based on Eq. (43), the caching node i's utility for offloading content f is formalized as follows:

$$\begin{aligned} u'_{if} &= p'_{if} - \phi_{if} \\ &= \sum_{l=1}^{L} \left(\delta'_{if}(l) x'_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) \\ &\quad + U^{-i}_{\mathcal{H}_f}(\mathcal{X}', \mathcal{Q}) - \phi_{if} \\ &= \sum_{l=1}^{L} \left(\delta'_{if}(l) x'_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) \\ &\quad + \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}'(l), \mathcal{Q}(l)) - x'_{if}(l)(\delta'_{if}(l) - B'_{if}(l))) - \phi_{if}. \end{aligned}$$

Therefore, the difference of each caching node's utility in set $\mathcal{H}$ after submitting the untruthful bid and the truthful bid can be calculated as follows:

$$\begin{aligned} &\Delta u_{if} = u'_{if} - u_{if} \\ &= \sum_{l=1}^{L} \left(\delta'_{if}(l) x'_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) \\ &\quad + \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}'(l), \mathcal{Q}(l)) - x'_{if}(l)(\delta'_{if}(l) - B'_{if}(l))) - \phi_{if} \\ &\quad - \sum_{l=1}^{L} \left(\delta_{if}(l) x_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) - U^{-i}_{\mathcal{H}_f}(\mathcal{X}, \mathcal{Q}) \\ &\quad + \phi_{if} \\ &= \sum_{l=1}^{L} \left(\delta'_{if}(l) x'_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) \\ &\quad + \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}'(l), \mathcal{Q}(l)) - x'_{if}(l)(\delta'_{if}(l) - B'_{if}(l))) \\ &\quad - \sum_{l=1}^{L} \left(\delta_{if}(l) x_{if}(l) - U_{\mathcal{H}(l)|(x_{if}\equiv 0)}(\mathcal{X}(l), \mathcal{Q}(l)) \right) \\ &\quad - \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) - x_{if}(l)(\delta_{if}(l) - B_{if}(l))) \\ &= \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}'(l), \mathcal{Q}(l) + x'_{if}(l) B'_{if}(l)) \\ &\quad - \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) + x_{if}(l) B_{if}(l)) \end{aligned}$$

Since $(\mathcal{X}(l), \mathcal{Q}(l))$ is the optimal solution that can maximize the saving cost function Eq. (12), we can get:

$$\begin{aligned} &\sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}'(l), \mathcal{Q}(l) + x'_{if}(l) B'_{if}(l)) \\ &\leq \sum_{l=1}^{L} (U_{\mathcal{H}(l)}(\mathcal{X}(l), \mathcal{Q}(l)) + x_{if}(l) B_{if}(l)). \end{aligned}$$

Therefore, $\Delta u_{if} \leq 0$. In other words, caching nodes cannot increase their utility from untruthful bids. □

TABLE II
PARAMETERS OF REAL MOBILITY TRACES

Trace	MIT Reality	Infocom 06
Duration (days)	299	4
The number of nodes	97	76
Granularity (seconds)	300	120
Device	Nokia 6600	iMote
Network type	Bluetooth	Bluetooth

VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed method and exploit the impact of different parameters.

A. Simulation Settings

The experiments use two real mobility traces: the MIT Reality trace [40] and the Infocom 06 trace [41]. The MIT Reality trace includes contact records of 299 days with 97 nodes, and the Infocom 06 trace includes contact records of 4 days with 76 nodes. The details of these two real traces are shown in TABLE II.

In the experiments, we consider that 15 contents are requested by different nodes at the same time, the size of the contents are within the range of $[100, 160]MB$, and the nodes' initial cache capacities are within the range of $[500, 600]MB$. For each node, we assume that the expected prices are within the range of $[0.01, 0.05]$ monetary units/MB, and the CSP's unit cost of traffic through cellular networks is 0.2 monetary units/MB. The transmission power of nodes is uniformly distributed in $[1, 2]W$, and the White Gaussian Noise power is $-100dBm$. In the proposed IDQNM, the minibatch and maximum replay memory sizes are set to 20 and 500, respectively. The learning rate ε is set as 0.001 and the discount factor of reward φ is set as 0.9. For the ϵ-greedy policy, we set $\epsilon = 1.0$ initially and let it decrease by a decay coefficient 0.099 over the time-slots until it reaches 0.1. Furthermore, the Sum of the Squared Errors (SSE) is used to fit the Pareto distribution of each node pair according to the historical contact records, which is expressed as:

$$SSE_{ij} = \sum_{k=1}^{\varpi_{ij}} (z_k - \hat{z}_k)^2, \tag{44}$$

where ϖ_{ij} is the number of contacts of node pair i and j, z_k and $\hat{z}_k$ are the actual duration and the fitted duration of node pair i and j at the kth contact, respectively. The smaller the value of SSE_{ij}, the better the fitting performance.

For performance comparison, we introduce the following three benchmark methods:

1) Incentive-driven and Decay-based Method (IDM) [12]: In the IDM, the CSP greedily selects caching nodes for each content at the beginning of each time slot, once a node is selected as the caching node for a content, the selection probability of its neighbors and neighbors' neighbors will be reduced by using a decay factor. If a node's cache capacity is not enough, it will

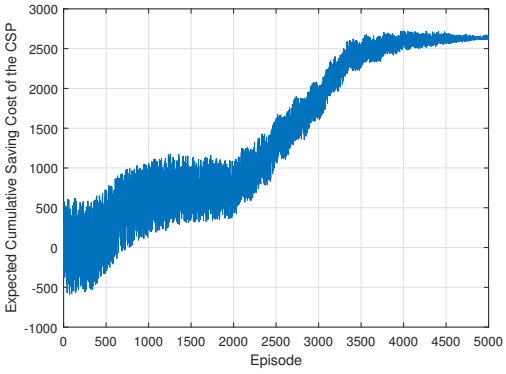


Fig. 3. Convergence of the proposed DQN-based content caching method.

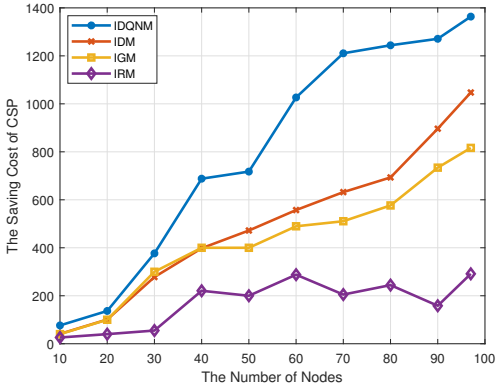


Fig. 4. Performance comparison in terms of the CSP's saving cost in the MIT Reality trace with different number of nodes.

replace the local cached content that has the minimum marginal contribution at the current time slot according to Eq. (40).

2) Incentive-driven and Greedy-based Method (IGM): In the IGM, the CSP selects the caching node set greedily according to the marginal contribution at each time slot. If a node's cache capacity is not enough, it will do the same as IDM.
3) Incentive-driven and Random-based Method (IRM): In the IRM, the CSP selects the same number of caching nodes as IDQNM at each time slot. If a node's cache capacity is not enough, it will randomly replace a local cached content that the caching node has no interest in.

For fairness, the payment rule of the above three methods is the same as that in IDQNM, which has been shown in algorithm 2. According to the optimization objective, we use the saving cost of the CSP and the offloading rate as the performance metrics. The offloading rate denotes the ratio of cellular traffic offloaded by D2D communications, which can be calculated as $\sum_{l=1}^{L}\tau_2(l)/\sum_{l=1}^{L}\tau_1(l)$, where $\tau_1(l)$ and $\tau_2(l)$ denote the total cellular traffic that should be transmitted by the CSP before selecting caching nodes at time slot l, and the amount of traffic transmitted via D2D communications after selecting caching nodes at time slot l, respectively.

B. Performance Comparison

In this part, the convergence performance of IDQNM is first illustrated. Then, the performance of IDQNM is compared with the three benchmark methods in terms of the CSP's saving cost and the offloading rate.

Fig. 3 shows the convergence performance of the proposed IDQNM in the MIT Reality trace. As shown in Fig. 3, in the proposed method, the expected cumulative saving cost of the CSP in each episode increases as the interaction between the agent (CSP) and the system environment continues, and the efficient data offloading strategy can be successfully learned without any prior-knowledge. Furthermore, it can be found that the expected cumulative saving cost of the CSP is relatively stable after about 3500 episodes.

Fig. 4 and Fig. 5 show the performance comparison in terms of the CSP's saving cost in the MIT Reality and Infocom 06 traces with different number of nodes, respectively. Considering that the contact frequency in the MIT Reality trace is much lower than that in the Infocom 06 trace, the maximum tolerant delay of content should be different. The maximum tolerant delay is set as $T_f^0 = [1, 3]$ days and $T_f^0 = [5, 10]$ hours in the MIT Reality and Infocom 06 traces, respectively.

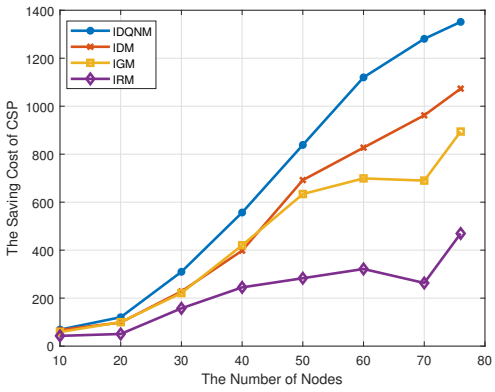


Fig. 5. Performance comparison in terms of the CSP's saving cost in the Infocom 06 trace with different number of nodes.

Fig. 4 illustrates the performance comparison in terms of

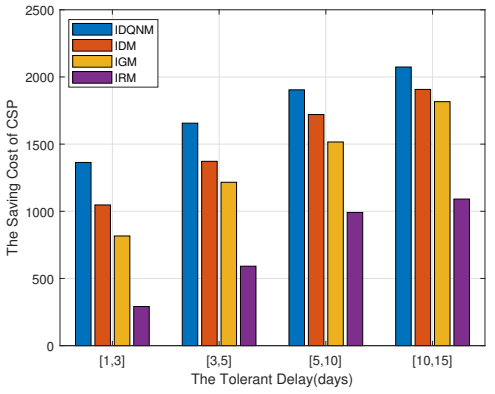


Fig. 6. Performance comparison in terms of the CSP's saving cost in the MIT Reality trace with different tolerant delay.

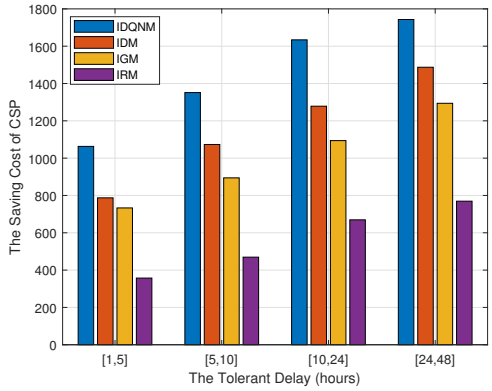


Fig. 7. Performance comparison in terms of the CSP's saving cost in the Infocom 06 trace with different tolerant delay.

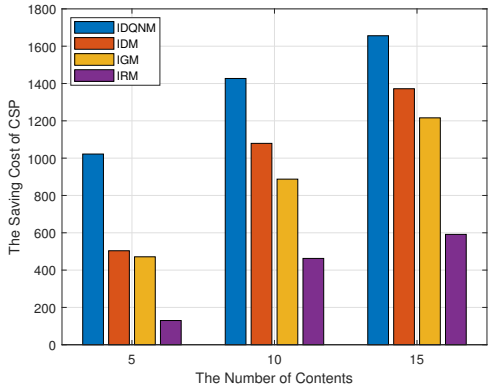


Fig. 8. Performance comparison in terms of the CSP's saving cost in the MIT Reality trace with different number of contents.

the CSP's saving cost in the MIT Reality trace. As shown in Fig. 4, with the increase of the number of nodes, the CSP's saving cost increases continually, and the proposed IDQNM outperforms other methods greatly, especially when the number of nodes is larger. The main reason for this is that more requests can be served by caching nodes as the number of nodes increases, and IDQNM can get the effective selected caching node set and content caching strategy from the long-term perspective in D2D offloading. Furthermore, IDM performs much better than IGM and IRM. The main reason is that compared with IGM, IDM adds a decay factor to update the offloading potential of each node, and selects far apart nodes with higher offloading potential and less payment as caching nodes, so more traffic can be offloaded through D2D communications in IDM. As expected, IGM performs better than IRM, and IRM performs the worst.

Fig. 5 illustrates the performance comparison in terms of the CSP's saving cost in the Infocom 06 trace. It can be found that IDQNM also outperforms other methods greatly as the number of nodes increases. IRM performs much better in the Infocom 06 trace than that in the MIT Reality trace. This is because in the Infocom 06 trace, there are more frequent contacts among mobile nodes, and the caching nodes have higher probabilities to contact other requested nodes, thus more contents will be delivered via D2D communications.

Fig. 6 and Fig. 7 show the performance comparison of different methods in terms of the CSP's saving cost in the two traces with different tolerant delay, respectively. Fig. 6 illustrates the performance comparison of different methods in the MIT Reality trace. As shown in Fig. 6, IDQNM can significantly increase the saving cost of the CSP, and performs best when the tolerant delay increases. This is because with the increase of the tolerant delay, mobile nodes are more likely to contact others within the deadline, which means more traffic can be offloaded repeatedly through D2D communications. Similar to the results above, as expected, IDM outperforms IGM and IRM, and IRM performs worst. Fig. 7 illustrates the performance comparison of different methods in the Infocom 06 trace. As shown in Fig. 7, IDQNM also performs best when the tolerant delay is different, especially when the tolerant

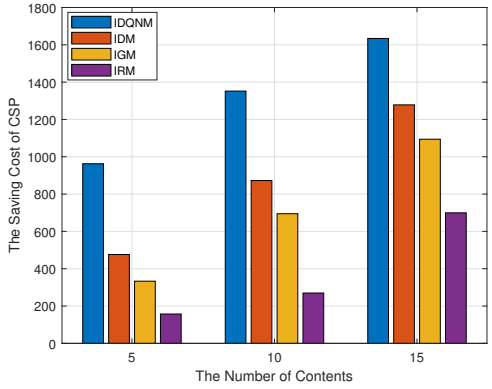


Fig. 9. Performance comparison in terms of the CSP's saving cost in the Infocom 06 trace with different number of contents.

delay is larger. Meanwhile, it can be seen that, whether in the MIT Reality trace or the Infocom 06 trace, as the tolerant delay increases, IDM outperforms IGM and IRM.

Fig. 8 and Fig. 9 illustrate the performance comparison of different methods in terms of the CSP's saving cost in the two traces with different number of contents, respectively. It can be found that IDQNM also outperforms other baseline methods greatly in the two traces as the number of contents increases. This is because when the number of contents increases, more contents are requested in the network, and so the effective selected caching node set and designed content caching method by IDQNM, from the long-term perspective, can maximize the saving cost of the CSP. Meanwhile, IDM can also increase the saving cost of the CSP significantly, especially when the number of contents is larger.

Fig. 10 and Fig. 11 show the performance comparison in terms of the offloading rate in the MIT Reality and Infocom 06 traces with different number of nodes, respectively. The saving cost of the CSP is closely related to the offloading rate. If the offloading rate is higher, it means that more traffic can be offloaded via D2D communications, then the CSP can save more cost. In Fig. 10, the tolerant delay is set as $T_f^0 = [1, 3]$ days. Similar to the results in Figs. 4 and 5, it can be found that the offloading rate of the proposed IDQNM is

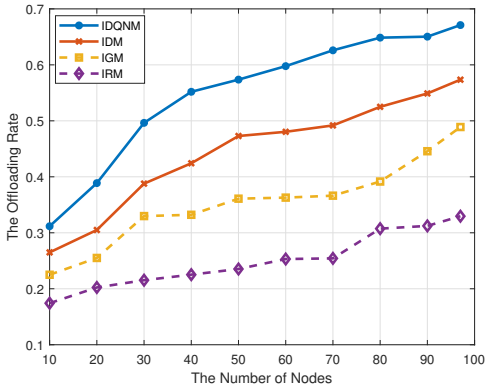


Fig. 10. Performance comparison in terms of the offloading rate in the MIT Reality trace with different number of contents.

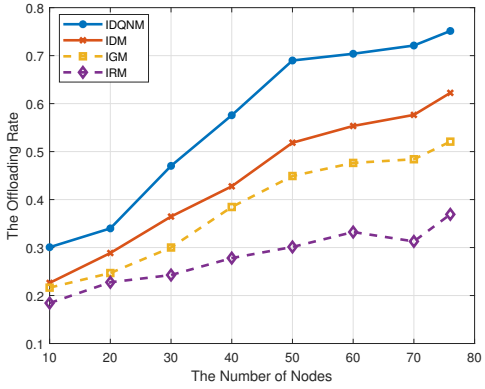


Fig. 11. Performance comparison in terms of the offloading rate in the Infocom 06 trace with different number of contents.

much higher than that of the other three benchmark methods, especially when the number of nodes increases. As expected, the performance of IDM is slightly better than IGM, and IRM has the lowest offloading rate since its caching nodes are randomly selected. In Fig. 11, the tolerant delay is set as $T_f^0 = [5, 10]$ hours. It can be found that the proposed IDQNM also performs best, and IRM performs worst. Therefore, it is proved that our proposed IDQNM performs best in terms of the offloading rate in both traces.

To summarize, our proposed IDQNM performs best in both traces under different scenarios. Therefore, we demonstrate that the proposed IDQNM is effective under different scenarios.

C. Evaluation of Individual Rationality and Truthfulness

In this part, we verify the individual rationality and truthfulness of the proposed payment determination in IDQNM. We verify the individual rationality by comparing the payment of each caching node with the corresponding true value of its cost in D2D communications. Furthermore, we verify the truthfulness by selecting a caching node randomly and allowing it to submit an untruthful bid which is not equal to the true value of its cost in D2D communications.

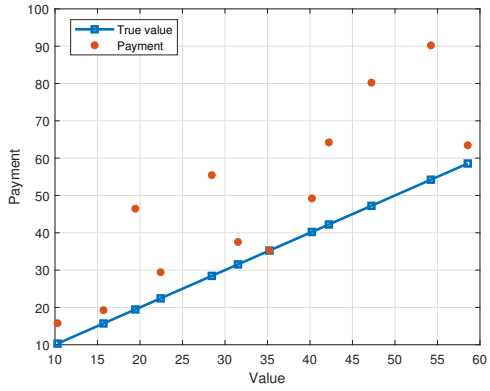


Fig. 12. Verification of the individual rationality

Fig. 12 shows the individual rationality of the proposed payment determination in the Infocom 06 trace. It can be found that 12 nodes are selected as the caching nodes by the proposed IDQNM. We assume that each caching node submits a truthful bid, the red dots represent the payment of each caching node, and the blue dots on the solid line indicate the corresponding true values of cost in D2D communications. Then, it can be found that the payment of each caching node is higher than the true value of its cost in D2D communications, which means that each caching node can get a positive reward when its bid is truthful. Therefore, we verify that the proposed payment determination can guarantee the individual rationality of caching nodes in IDQNM.

In Fig. 13, we select a caching node randomly in the Infocom 06 trace according to IDQNM, and the true value of its cost in D2D communications is 42.23. Then, it can be found that when the claimed bid of the selected node is less than the true value of its cost in D2D communications, it will not be willing to be selected as a caching node, and when its claimed bid is too large, the CSP will not be willing to choose it as a caching node from the economic perspective. Only when the claimed bid of the selected node is in the range of [43, 45], this node can be selected as the caching node, and its payment is always 64.23 even its claimed bid increases, thus its payoff is a constant that is equal to the payment minus its true value. Therefore, it can be proved that the caching nodes cannot get higher payments from untruthful bids.

To summarize, we prove that the proposed payment determination can guarantee the individual rationality and truthfulness properties of the proposed IDQNM, which can encourage each node to submit a truthful bid to maximize the saving cost of the CSP and maximize the utility of itself.

VIII. Conclusion

In this paper, the content caching strategy and incentive mechanism have been jointly considered to improve the performance of D2D offloading. A novel Incentive-driven and DQN based Method, named IDQNM is proposed, in which reverse auction is employed as the incentive mechanism. Then, the incentive-driven D2D offloading and content caching process is modeled as an INLP problem, aiming to maximize the

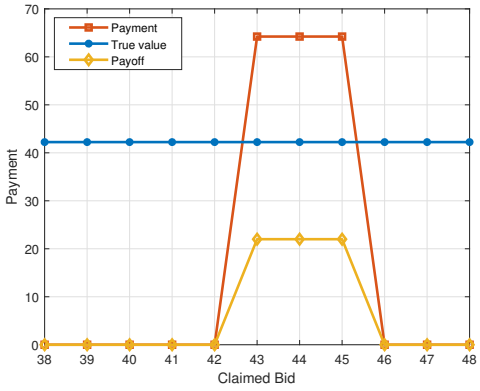


Fig. 13. Verification of the truthfulness

saving cost of the CSP. To solve the optimization problem, a DQN-based content caching method is proposed to get the approximate optimal solution. Moreover, a VCG-based payment rule is proposed, which can guarantee nodes' individual rationality and truthfulness properties of the proposed IDQNM. Real trace-driven simulation results demonstrate that IDQNM outperforms the other baseline methods greatly in terms of the CSP's saving cost and the offloading rate under different scenarios.

Appendix

In this part, we introduce the details regarding how to obtain the expression of Eq. (6).

Similar to studies in [12], [37], we assume that the inter-contact time of node pairs in real mobility traces follows an exponential distribution. Here, T_{ij}^{inter} is used to denote the inter-contact time of a node pair i and j, i.e. $T_{ij}^{inter} \sim \text{Exp}\left(\lambda_{ij}\right)$, and λ_{ij} is given as:

$$\lambda_{ij} = \frac{\varpi_{ij}}{\sum_{m=1}^{\varpi_{ij}} ICO_{ij}^{m}}, \tag{45}$$

where ICO_{ij}^{m} denotes the inter-contact time samples of the node pair i and j, and ϖ_{ij} represents the number of contacts between them. In general, a content may not be completely delivered during one contact between a pair of nodes, thus we need to consider the probability of multiple contacts within the time constraint between nodes i and j. Since each contact between nodes i and j is independently distributed, we have $\mathbb{T}_{ij} \sim \Gamma\left(\varpi_{ij}, \lambda_{ij}\right)$, where $\mathbb{T}_{ij} = \sum_{m=1}^{\varpi_{ij}} T_{ij}^{inter}$. The probability density functions (PDF) of $\mathbb{T}_{ij}$ is:

$$f^{\mathbb{T}_{ij}}(t) = f\left(t; \varpi_{ij}, \lambda_{ij}\right) = \frac{t^{\varpi_{ij}-1}}{\lambda_{ij}^{-\varpi_{ij}}\Gamma\left(\varpi_{ij}\right)} e^{-t\lambda_{ij}}. \tag{46}$$

Then, the probability that nodes i and j have ϖ_{ij} contacts within the time constraint $P\left(\mathbb{T}_{ij} \le \mathcal{T}_f(l)\right)$ is expressed as:

$$\begin{aligned} P\left(\mathbb{T}_{ij} \le \mathcal{T}_f(l)\right) &= \int_{\mathcal{T}_f(l)}^{+\infty} f\left(t; \varpi_{ij}+1, \lambda_{ij}\right) \cdot dt \\ &= \sum_{k=0}^{\varpi_{ij}} \frac{\left(\lambda_{ij}\mathcal{T}_f(l)\right)^k \cdot e^{-\lambda_{ij}\mathcal{T}_f(l)}}{k!}. \end{aligned} \tag{47}$$

The contact duration is modeled as the Pareto distribution, thus the size of the content that can be delivered during a contact between nodes i and j also follows the Pareto distribution. Let G_{ij}^{k} denote the size of data transmitted during a contact between a node pair i and j, where k represents the kth contact between them. According to the previous definition, $G_{ij}^{k}(k = 1, 2, ..., \varpi_{ij})$ follows the Pareto distribution with scale parameter β_{ij} and shape parameter α_{ij}, that is $G_{ij}^{k} \sim$ Pareto $(\alpha_{ij}, \beta_{ij})$, where β_{ij} is the minimum possible value of G_{ij}^{k}. α_{ij} and β_{ij} can be calculated by fitting the historical contact records between node i and j. $\mathcal{G}_{ij} = \sum_{k=1}^{\varpi_{ij}} G_{ij}^{k}$ is used to denote the total amount of traffic delivered through ϖ_{ij} times opportunistic communication. Then, the probability that node j's request of content f can be delivered with the number of contacts ϖ_{ij} by node i can be represented as $P\left(\mathcal{G}_{ij} \ge q_{jf}(l)\right)$. However, it is difficult to approximate the PDF of $\mathcal{G}_{ij}$ by a stable distribution since $\mathcal{G}_{ij}$ is the sum of an arbitrary number of random variables G_{ij}^{k}. Thus, an approximate algorithm is used to estimate $P\left(\mathcal{G}_{ij} \ge q_{jf}(l)\right)$. Let w denote the maximum value of $G_{ij}^{k}(k = \{1, \ldots, \varpi_{ij}\})$, and $\Delta = \mathcal{G}_{ij}/w$, then $P\left(\mathcal{G}_{ij} \ge q_{jf}(l)\right)$ can be approximated by Δ [42]:

$$P\left(\mathcal{G}_{ij} \ge q_{jf}(l)\right) \approx 1 - \left(1 - \left(\frac{\beta_{ij}\bar{\Delta}}{q_{jf}(l)}\right)^{\alpha_{ij}}\right)^{\varpi_{ij}}, \tag{48}$$

where $\bar{\Delta}$ represents the expectation of Δ, which can be formulated as follows:

$$\bar{\Delta} = \begin{cases} \frac{1-\varpi_{ij}B\left(\varpi_{ij}, \alpha_{ij}^{-1}\right)}{1-\alpha_{ij}}, & \alpha_{ij} \ne 1 \\ \sum_{i=1}^{\varpi_{ij}} i^{-1}, & \alpha_{ij} = 1. \end{cases} \tag{49}$$

Given node j's requested size of content f and the tolerant delay at time slot l, the probability that $q_{jf}(l)$ can be successfully delivered via D2D communications within the tolerant delay $\mathcal{T}_f(l)$ can be obtained. Let $P_{ijf}\left(\mathcal{T}_f(l), q_{jf}(l)\right)$ represent the probability that node j's request can be satisfied by node i within $\mathcal{T}_f(l)$, and $\varpi_{ijf}^{max}(l) = \left\lceil \frac{q_{jf}(l)}{\beta_{ij}} \right\rceil$ represent the maximum possible number of contacts between nodes i and j during time slot l, then according to Eq. (6), $P_{ijf}\left(\mathcal{T}_f(l), q_{jf}(l)\right)$ can be calculated as follows:

$$\begin{aligned} &P_{ijf}\left(\mathcal{T}_f(l), q_{jf}(l)\right) \\ &= \sum_{k=1}^{\varpi_{ijf}^{max}(l)} \widehat{P}_{k-1} \cdot P\left(\mathcal{G}_{ij}^{k} \ge q_{jf}(l)\right) \cdot P(\mathbb{T}_{ij}^{k} \le \mathcal{T}_f(l) - T_{ijf}^{trans}(l)) \\ &= \sum_{k=1}^{\varpi_{ijf}^{max}(l)} \left[\widehat{P}_{k-1} \cdot \left(1 - \left(1 - \left(\frac{\beta_{ij}\bar{\Delta}}{q_{jf}(l)}\right)^{\alpha_{ij}}\right)^{k}\right)\right. \\ &\quad \left. \cdot \sum_{k=0}^{\varpi_{ij}} \frac{\left(\lambda_{ij}\mathcal{T}_f(l)\right)^k \cdot e^{-\lambda_{ij}\mathcal{T}_f(l)}}{k!}\right]. \end{aligned} \tag{50}$$

References

[1] D. Xu, Y. Li, X. Chen, J. Li, P. Hui, S. Chen, and J. Crowcroft, "A Survey of Opportunistic Offloading," *IEEE Commun. Surveys Tut.*, vol. 20, no. 3, pp. 2198-2236, 2018.

[2] X. Zhang, and Q. Zhu, "D2D Offloading for Statistical QoS Provisionings Over 5G Multimedia Mobile Wireless Networks," *in Proc. IEEE INFOCOM*, pp. 82-90, 2019.

[3] E. Hossain, and M. Hasan, "5G cellular: key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11-21, June 2015.

[4] Forecast G M D T. Cisco visual networking index: global mobile data traffic forecast update, 2017-2022. Update, 2019.

[5] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-Efficiency Oriented Traffic Offloading in Wireless Networks: A Brief Survey and a Learning Approach for Heterogeneous Cellular Networks." *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627-640, Apr. 2015.

[6] H. Zhou, H. Wang, X. Li, and V. C. M. Leung, "A Survey on Mobile Data Ofoading Technologies," *IEEE Access*, vol. 6, no. 1, pp. 5101-5111, Jan. 2018.

[7] H. Zhou, X. Chen, S. He, J. Chen, and J. Wu, "DRAIM: A Novel Delay-constraint and Reverse Auction-based Incentive Mechanism for WiFi Offloading," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 4, pp. 711-722, 2020.

[8] F. Rebecchi, M. D. Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: a survey," *IEEE Commun. Surveys Tut.*, vol. 17, no. 2, pp. 580-603, 2nd Quart. 2015.

[9] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, "Device-Enhanced MEC: Multi-Access Edge Computing (MEC) Aided by End Device Computation and Caching: A Survey," *IEEE Access*, vol. 7, pp. 166079-166108, 2019.

[10] D. Chatzopoulos, C. Bermejo, E. UI Haq, Y. Li, and P. Hui, "D2D Task Offloading: A Dataset-Based Q&A," *IEEE Commun. Mag.,* vol. 57, no. 2, pp. 102-107, Feb. 2019.

[11] Y. Li, D. Jin, Z. Wang, P. Hui, L. Zeng, and S. Chen, "Multiple Mobile Data Offloading Through Disruption Tolerant Networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1579-1596, Jul. 2014.

[12] H. Zhou, X. Chen, S. He, C. Zhu, and Victor C. M. Leung. "Freshness-aware Seed Selection for Offloading Cellular Traffic through Opportunistic Mobile Networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2658-2669, April 2020.

[13] B. Han, P. Hui, V. S. A. Kumar, M. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 821-834, 2012.

[14] W. Sun, J. Liu, Y. Yue, and Y. Jiang, "Social-Aware Incentive Mechanisms for D2D Resource Sharing in IIoT," *IEEE Trans. Ind Informal.*, vol. 16, no. 8, pp. 5517-5526, Aug. 2020.

[15] Y. Pan, C. Pan, Z. Yang, M. Chen, and J. Wang, "A Caching Strategy Towards Maximal D2D Assisted Offloading Gain," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2489-2504, 1 Nov. 2020.

[16] Y. Zhang, J. Li, Y. Li, D. Xu, M. Ahmed, and Y. Li, "Cellular Traffic Offloading via Link Prediction in Opportunistic Networks," *IEEE Access*, vol. 7, pp. 39244-39252, 2019.

[17] C. Yang, and R. Stoleru, "CEO: Cost-Aware Energy Efficient Mobile Data Offloading via Opportunistic Communication," *in Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, pp. 548-554, 2020.

[18] S. Yu, B. Dab, Z. Movahedi, R. Langar, and L. Wang, "A Socially-Aware Hybrid Computation Offloading Framework for Multi-Access Edge Computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1247-1259, Jun. 2020.

[19] X. Zhang, P. Huang, L. Guo, and Y. Fang, "Social-Aware Energy-Efficient Data Offloading With Strong Stability," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1515-1528, Aug. 2019.

[20] Y. Zhao, W. Song, and Z. Han, "Social-Aware Data Dissemination via Device-to-Device Communications: Fusing Social and Mobile Networks with Incentive Constraints," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 489-502, May-Jun. 2019.

[21] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-Agent Reinforcement Learning Based Cooperative Content Caching for Mobile Edge Networks," *IEEE Access*, vol. 7, pp. 61856-61867, 2019.

[22] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks," *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 247-257, Jan. 2020.

[23] S. Sakib, T. Tazrin, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "An Efficient and Light-weight Predictive Channel Assignment Scheme for Multi-Band B5G Enabled Massive IoT: A Deep Learning Approach," *IEEE Internet of Things J.,* to be published, doi: 10.1109/JIOT.2020.3032516.

[24] X. Wang, R. Li, C. Wang, X. Li, T. Taleb, and V. C. M. Leung, "Attention-Weighted Federated Deep Reinforcement Learning for Device-to-Device Assisted Heterogeneous Collaborative Edge Caching," *IEEE J. Sel. Areas Commun.,* to be published, doi: 10.1109/JSAC.2020.3036946.

[25] C. Wang, S. Wang, D. Li, X. Wang, X. Li, and V. C. M. Leung, "Q-Learning Based Edge Caching Optimization for D2D Enabled Hierarchical Wireless Networks," *in Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, pp. 55-63, Oct, 2018.

[26] H. Shah-Mansouri, V. W. S. Wong, and J. Huang, "An Incentive Framework for Mobile Data Offloading Market Under Price Competition," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 2983-2999, Nov. 2017.

[27] R. Chattopadhyay, and C. Tham, "Fully and Partially Distributed Incentive Mechanism for a Mobile Edge Computing Network," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2020.3003079.

[28] Y. Chen, S. He, F. Hou, Z. Shi, and J. Chen, "An Efficient Incentive Mechanism for Device-to-Device Multicast Communication in Cellular Networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 7922-7935, 2018.

[29] J. Du, E. Gelenbe, C. Jiang, Z. Han, and Y. Ren, "Auction-Based Data Transaction in Mobile Networks: Data Allocation Design and Performance Analysis," *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1040-1055, 2020.

[30] S. Paris, F. Martignon, I. Filippini, and L. Chen, "An Efficient Auction-based Mechanism for Mobile Data Offloading," *IEEE Trans. Mobile Comput.*, vol. 14, no. 8, pp. 1573-1586, 2015.

[31] W. Song, and Y. Zhao, "A Randomized Reverse Auction for Cost-Constrained D2D Content Distribution," *in Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec, 2016, pp. 1-6.

[32] P. Li, S. Guo, and, I. Stojmenovic, "A Truthful Double Auction for Device-to-Device Communications in Cellular Networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 71-81, Jan. 2016.

[33] T. Zhang, X. Fang, Y. Liu, G. Y. Li, and W. Xu, "D2D-Enabled Mobile User Edge Caching: A Multi-Winner Auction Approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12314-12328, Dec. 2019.

[34] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction Design and Analysis for SDN-Based Traffic Offloading in Hybrid Satellite-Terrestrial Networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202-2217, Oct. 2018.

[35] D. Liu, A. Hafid, and L. Khoukhi, "Multi-Item Auction Based Mechanism for Mobile Data Offloading: A Robust Optimization Approach," *IEEE Trans. Veh. Technol*, vol. 69, no. 4, pp. 4155-4168, Apr. 2020.

[36] J. Du, C. Jiang, E. Gelenbe, H. Zhang, Y. Ren, and T. Q. S. Quek, "Double Auction Mechanism Design for Video Caching in Heterogeneous Ultra-Dense Networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1669-1683, March 2019.

[37] Z. Lu, X. Sun, and T. La Porta, "Cooperative Data Offload in Opportunistic Networks: From Mobile Devices to Infrastructure," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3382-3395, Dec. 2017.

[38] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine Learning for 6G Wireless Networks: Carrying Forward Enhanced Bandwidth, Massive Access, and Ultrareliable/Low-Latency Service," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 122-134, Dec. 2020.

[39] Z. Ning, P. Dong, X. Wang, L. Guo, J. J. Rodrigues, X. Kong, J. Huang, and R. Y. Kwok "Deep Reinforcement Learning for Intelligent Internet of Vehicles: An Energy-Efficient Computational Offloading Scheme," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1060-1072, Dec. 2019.

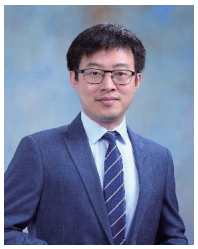
[40] A. Pentland, N. Eagle, and D. Lazer, "Inferring social network structure using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15274-15278, 2009.

[41] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "Crawdad data set cambridge/haggle (v. 2009-05-29)," 2009.

[42] I. V. Zaliapin, Y. Y. Kagan, and F. P. Schoenberg. "Approximating the Distribution of Pareto Sums". *Pure Appl. Geophys.*, vol. 162, nos. 6-7, pp. 1187-1228, 2005.

Huan Zhou (M'14) received his Ph. D. degree from the Department of Control Science and Engineering at Zhejiang University. He was a visiting scholar at the Temple University from Nov. 2012 to May, 2013, and a CSC supported postdoc fellow at the University of British Columbia from Nov. 2016 to Nov. 2017. Currently, he is a full professor at the College of Computer and Information Technology, China Three Gorges University. He was a Lead Guest Editor of Pervasive and Mobile Computing, TPC Chair of EAI BDTA 2020, Local Arrangement Chair of I-SPAN 2018, Special Session Chair of the 3rd International Conference on Internet of Vehicles (IOV 2016), and TPC member of IEEE Globecom, ICC, ICCCN, etc. He has published more than 50 research papers in some international journals and conferences, including IEEE JSAC, TPDS, TWC and so on. His research interests include Opportunistic Mobile Networks, VANETs, Mobile Data Offloading, and Mobile Edge Computing. He receives the Best Paper Award of I-SPAN 2014 and I-SPAN 2018, and is currently serving as an associate editor for IEEE ACCESS and EURASIP Journal on Wireless Communications and Networking.

Tong Wu received his B.S. Degree from Jincheng College of Sichuan University. Currently, he is a graduate student at the College of Computer Information and Technology, China Three Gorges University. His research interests include mobile edge caching and opportunistic mobile networks.



Haijun Zhang (M'13, SM'17) is currently a Full Professor and Associate Dean at University of Science and Technology Beijing, China. He was a Postdoctoral Research Fellow in Department of Electrical and Computer Engineering, the University of British Columbia (UBC), Canada. He serves/served as Track Co-Chair of WCNC 2020, Symposium Chair of Globecom'19, TPC Co-Chair of INFOCOM 2018 Workshop on Integrating Edge Computing, Caching, and Offloading in Next Generation Networks, and General Co-Chair of GameNets'16. He serves as an Editor of IEEE Transactions on Communications, IEEE Transactions on Network Science and Engineering, and IEEE Transactions on Vehicular Technology. He received the IEEE CSIM Technical Committee Best Journal Paper Award in 2018, IEEE ComSoc Young Author Best Paper Award in 2017, and IEEE ComSoc Asia-Pacific Best Young Researcher Award in 2019.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Services Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, IEEE ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.