Joint Design of User Association, Caching and Power Allocation for Delay Optimization in UAV-Enabled Networks

Gezahegn Abdissa Bayessa[1], Rong Chai[1(✉)], Yetmwork Gutema Lemu[2], and Qianbin Chen[1]

[1] Chongqing University of Posts and Telecommunications, Chongqing 400065, China
L202010006@stu.cqupt.edu.cn, {chairong,cqb}@cqupt.edu.cn
[2] Adama Science and Technology University, Adama, Ethiopia

Abstract. The rapid surge of multimedia and video applications poses challenges to the content-delivering service in wireless networks. In this paper, we study the proactive caching problem in unmanned aerial vehicle (UAV)-enabled networks, where a number of UAVs are deployed to offer content delivery service for user equipments (UEs). In order to acquire user request information, we first propose a bidirectional long short-term memory-based user request prediction algorithm. Then, based on the obtained user content requests, we examine the content fetching delay of users and formulate UAV deployment, content caching, and power allocation problems as an overall content fetching delay minimization problem. To solve the formulated optimization problem, we decouple it into two subproblems, namely, the UAV deployment and content caching subproblem, and the power allocation subproblem, and solve the two subproblems by using an alternate iteration-based algorithm. Specifically, we first design a modified K-means-based clustering scheme to group UEs into various clusters, and then develop a UAV deployment strategy for individual clusters by applying quadratic transformation and first-order Taylor expansion. A heuristic proactive content caching algorithm is further proposed for individual UAVs. Finally, the Lagrangian dual method is employed to solve the power allocation subproblem. Simulation results demonstrate the effectiveness of the proposed algorithms.

Keywords: Unmanned aerial vehicles (UAVs) · user clustering · UAV deployment · proactive content caching · power allocation

1 Introduction

The proliferation of multimedia and video applications brought rapid growth of data traffic, which in turn poses formidable challenges to the transmission performance of the traditional cellular systems [1]. In response, proactive caching technology which stores popular files at network edge has been proposed as a promising solution [2]. Recently, unmanned aerial vehicles (UAVs) have garnered

X. Cheng (Ed): BROADNETS 2024, LNICST 601, pp. 95–113, 2025.
https://doi.org/10.1007/978-3-031-81168-5_10

significant attention as a potential platform for proactive caching. Leveraging the flexible deployment and scalability of UAVs, UAV-aided caching is capable of improving the efficiency of the content delivery [3].

In recent years, content delivery problem has been widely studied for UAV-enabled networks [4–10]. The research works in [4,5] investigate a content caching algorithm to maximize content availability [4] or to minimize energy consumption [5]. In [6], the authors propose a long short-term memory (LSTM)-based caching algorithm to minimize the weighted delivery cost. To address the challenges associated with LSTM in capturing dependencies and adapting to patterns of sequential input data in both forward and backward directions, bidirectional long short-term memory (BiLSTM) have been employed in [7]. By considering UAV deployment, the research works in [8–10] propose a joint content placement scheme and UAV deployment strategy to minimize storage cost [8], maximize throughput [9] and maximize mean opinion score [10].

The research works in [11,12] jointly consider UAV deployment, cache placement and power allocation issues to minimize the average outage probability [11] or to maximize the hit probability [12]. User clustering and resource allocation are considered along with UAV deployment and cache placement strategy in [13–17]. The authors propose a K-means algorithm for user clusters to deploy UAVs, and design a resource allocation strategy to fulfill users' QoS requirements. The authors in [13] design a quality of experience (QoE) based user grouping scheme that finds optimal positions of UAVs and stores popular contents to maximize content delay index. In [14] the authors propose joint optimization of UAV deployment and resource allocation for UAV-aided relay systems to maximize the energy efficiency. The research work in [15] introduces joint UAV deployment and communication resource allocation to maximize the total long-term QoE of users in multiuser video streaming in UAV relay networks. The authors in [16] design joint deployment of aerial base stations (ABSs), user associations, and corresponding bandwidth allocations to minimize total downlink transmit power in cache-enabled wireless networks. Research work in [17] presents joint user clustering, UAV deployment, and resource allocation algorithms to minimize transmit power and improve overall spectral efficiency (SE).

Although content caching in UAV-enabled networks have been exploited in aforementioned research works, yet the existing research mainly design joint UAV deployment and caching strategy based on user locations and rarely considers the impact of user request distribution on the UAV deployment. On the other hand, recent studies assume perfect knowledge of users' content requests [4,5], which may not be practical. As the content caching strategy is designed proactively, the difficulty of acquiring explicit user demand information in advance poses a significant challenge in designing a joint strategy. Few existing solutions for unknown content popularity are mostly based on conventional predict-then-optimize schemes (predict the content popularity first and then optimize the cache policy) [6,8]. While it can achieve good performance when user demand

follows clear patterns, it is far less effective when explicit user demand information is not available. In this context, the existing content prediction schemes are not well-suited for addressing the challenges posed by unknown content request patterns. Thus, the problem of joint UAV deployment, content placement and power allocation with unknown user requests remains open for multi-UAV-assisted wireless networks.

In this paper, we consider the content delivery problem in a UAV-enabled network and jointly design UAV deployment, content caching, and power allocation strategy. Specifically, we first propose a bidirectional long short-term memory (BiLSTM)-based content request prediction algorithm. Then, based on the obtained user request information, we examine the overall delay incurred for content fetching in the network and formulate the joint UAV deployment, content caching, and power allocation problem as an overall delay minimization problem. To solve the formulated problem, we transform it into two subproblems and solve them respectively. Specifically, we first design a UAV deployment strategy through utilizing quadratic transformation and first-order Taylor expansion. Then, we design a heuristic proactive content caching algorithm for the UAVs. Based on the obtained local UAV deployment and content caching strategy, we then design a power allocation strategy by means of the Lagrangian dual method (Fig. 1).

2 System Model

2.1 Network Model

In this paper, we consider the content delivery service of UEs in a wireless network. Suppose that UEs have some content fetching requirements. As the content server is in general deployed at the core network which is far from the UEs, resulting in undesired transmission performance. To tackle this problem, we deploy a number of UAVs, which are equipped with certain cache capacity. The UAVs are capable of retrieving contents requested by users and store the contents locally in their own storage. By accessing the UAVs, UEs are able to fetching their requested contents directly from the UAVs instead of interacting with the core network. We denote the number of UAVs to be deployed as J and the number of UEs as I. Let UAV_j denote the j-th UAV, and UE_i denote the i-th UE, $1 \leq i \leq I$, $1 \leq j \leq J$.

To enable multi-user accessing on an individual UAV, we apply orthogonal frequency division multiple access (OFDMA) scheme which allows multiple UEs to access one UAV using orthogonal subcarriers. We denote B as subcarrier bandwidth.

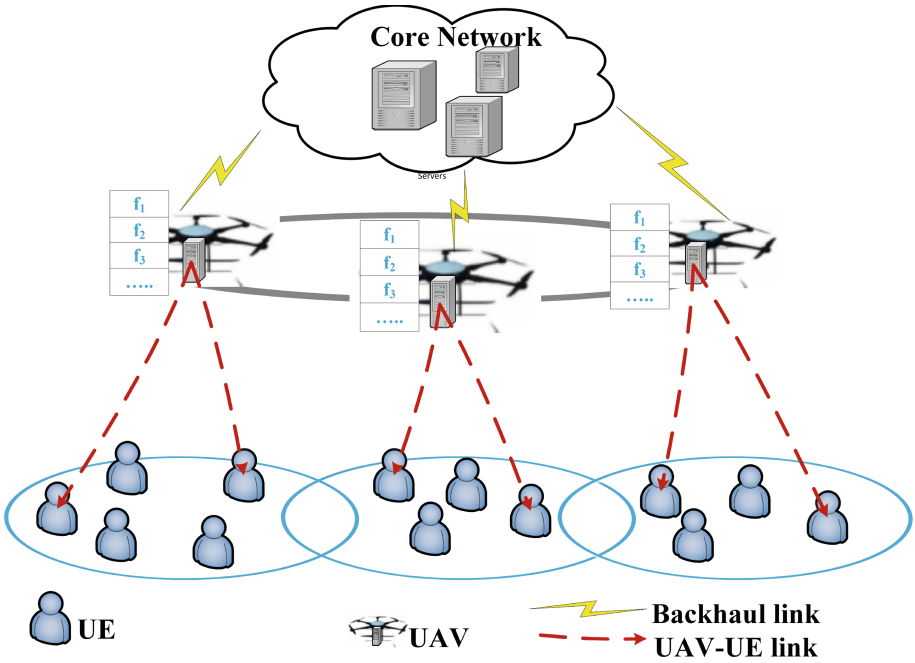

Fig. 1. The UAV-enabled network: a number of UAVs and several UEs.

Suppose that all the files are stored at the remote content server and each UAV may cache certain contents. Let F denote the total number of content files that UEs request, η_f denote the size of file f and ρ_j denote the cache capacity of UAV_j. We assume that each UE is allowed to associate with one UAV and denote UE association variable as $\xi_{i,j}$. If UE_i is associated with UAV_j, $\xi_{i,j} = 1$, otherwise $\xi_{i,j} = 0$.

Without loss of generality, we assume that the UAVs are deployed at a constant hovering altitude z. Let $\mathbf{q}_j = (x_j, y_j)$ denote the two-dimensional coordinate of UAV_j and $\mathbf{n}_i = (\bar{x}_i, \bar{y}_i)$ denote the coordinate of UE_i.

2.2 Communication Model

In this subsection, we discuss the channel model of UAV-UE links and then formulate the data rate of the transmission links. Assuming the UAV-UE links experience free-space path-loss. Let $h_{i,j}$ denote the channel coefficient of the link from UAV_j to UE_i, which can be expressed as

$$h_{i,j} = \frac{c}{4\pi f_0(\|\mathbf{q}_j - \mathbf{n}_i\|^2 + z^2)}, \tag{1}$$

where c is the speed of light and f_0 denotes the carrier frequency.

Let P_j denote the total transmit power of UAV_j, $\alpha_{i,j}$ denote the fraction of power allocated to UE_i when accessing UAV_j, $0 \leq \alpha_{i,j} \leq 1$. Denote R_i as the achievable data rate of UE_i, which can be computed as

$$R_{i,j} = B \log_2(1 + \frac{\alpha_{i,j} P_j |h_{i,j}|}{\sigma^2}). \tag{2}$$

3 Content Request Prediction

In the UAV-enabled content delivery system, the content fetching performance is expected to be improved by caching user requests. However, user content requests may vary over time, thus posing challenges to content caching in UAVs. To resolve this problem, we first predict user requests based on their historical request information. Specifically, for each user, we propose a BiLSTM-based algorithm to predict the probabilities for requesting all the files.

In a BiLSTM network, two hidden layers are connected to the same output layer and pass the learned prediction history forward and backward. We denote $\mathbf{h}_t^{\text{f}}$ and $\mathbf{h}_t^{\text{b}}$ respectively as the forward and backward hidden states of the BiLSTM architecture at time slot t. The final output of the BiLSTM model is a concatenation of the forward and backward hidden state sequences. Let $\hat{\mathbf{y}}_t$ denote the output of the BiLSTM model, which can be given as

$$\hat{\mathbf{y}}_t = \phi\left(\mathbf{W}_{hy}\left[\mathbf{h}_t^{\text{f}}; \mathbf{h}_t^{\text{b}}\right] + \mathbf{b}_y\right), \tag{3}$$

where $[\mathbf{h}_t^{\text{f}}; \mathbf{h}_t^{\text{b}}]$ denotes the concatenation of forward and backward hidden states, $\mathbf{W}_{hy}$ is the weight matrix for the output layer, $\mathbf{b}_y$ is the bias vector for the output layer, and $\phi(\cdot)$ is the activation function.

Let $\gamma_{i,f,t}$ denote the content request identifier of UE_i at time slot t. We set $\gamma_{i,f,t} = 1$, if UE_i requires content f at time slot t, otherwise, $\gamma_{i,f,t} = 0$, $1 \leq i \leq I$, $1 \leq f \leq F$. Let $x_{i,f,t}$ denote the number of requests of UE_i for content f until time slot t, we obtain $x_{i,f,t} = \sum_{t_1=1}^{t} \gamma_{i,f,t_1}$. Denote $\mathbf{x}_{i,t}$ as the content request vector of UE_i at time slot t, which can be expressed as

$$\mathbf{x}_{i,t} = \left[x_{i,1,t}, \cdots, x_{i,f,t}, \cdots, x_{i,F,t}\right]. \tag{4}$$

We set the input of the BiLSTM-based prediction model of UE_i at time slot t as $\mathbf{x}_{i,t}$. Given input sequence $\mathbf{x}_{i,t}$, we train the BiLSTM model based on the historical request data of UE_i. Then, the output of the BiLSTM model at $t+1$ which is the predicted request of UE_i, can be expressed as

$$\hat{\mathbf{y}}_{i,t+1} = \left[\hat{y}_{i,1,t+1}, \cdots, \hat{y}_{i,f,t+1}, \cdots, \hat{y}_{i,F,t+1}\right], \tag{5}$$

where $\hat{y}_{i,f,t+1}$ is the probability that UE_i requests content f at time slot $t+1$. To describe user content request we introduce $\gamma_{i,f}$. Based on $\hat{\mathbf{y}}_{i,t+1}$, we obtain $\gamma_{i,f}$. Specifically, if $f^* = \arg\max_{\forall f} \hat{y}_{i,f,t+1}$, we set $\gamma_{i,f^*} = 1$, otherwise, $\gamma_{i,f^*} = 0$.

4 Problem Formulation

In this paper, we jointly design the UAV deployment, content caching and power allocation strategy so as to minimize overall delay.

4.1 Objective Function Formulation

Considering that content requests at UAVs follow a Poisson distribution and request processing at the UAVs follows an exponential distribution, we model the content request processing at UAVs using an M/G/1 queuing model. Let λ_j^{q} and μ_j^{q} denote the request rate and service rate at UAV$_j$, respectively, D is the overall delay incurred for fetching requested contents, which can be given by,

$$D = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{f=1}^{F}\xi_{i,j}\gamma_{i,f}\left(\frac{\eta_f}{R_{i,j}} + \frac{1}{\mu_j^{\mathrm{q}} - \lambda_j^{\mathrm{q}}}\right) + \sum_{j=1}^{J}\sum_{f=1}^{F}(1-\delta_{j,f})\eta_f D_j^{\mathrm{s}}, \tag{6}$$

$\delta_{j,f}$ is the content caching variable of UAV$_j$. We set $\delta_{j,f} = 1$, if UAV$_j$ caches content f, otherwise, $\delta_{j,f} = 0$. D_j^{s} is the backhaul delay of UAV$_j$ for fetching unit content from the content server. Without loss of generality, we consider D_j^{s} as a constant.

4.2 Optimization Problem Formulation

We formulate the joint UAV deployment, content caching and power allocation as a delay minimization problem, which is given as follows

$$\begin{aligned}
\mathrm{P1}: \min_{\{\xi_{i,j}\},\{\mathbf{q}_j\},\{\delta_{j,f}\},\{\alpha_{i,j}\}} \quad & D \\
\text{s.t. } \mathrm{C1}: & \sum_{j=1}^{J}\xi_{i,j} \le 1, \forall i, \\
\mathrm{C2}: & \sum_{i=1}^{I}\xi_{i,j} \le N_0, \forall j, \\
\mathrm{C3}: & 0 \le x_j \le x_{\max}, \forall j, \\
\mathrm{C4}: & 0 \le y_j \le y_{\max}, \forall j, \\
\mathrm{C5}: & \sum_{f=1}^{F}\eta_f\delta_{j,f} \le \rho_j, \forall j, \\
\mathrm{C6}: & \sum_{i=1}^{I}\alpha_{i,j} \le 1, \forall j, \\
\mathrm{C7}: & R_{i,j} \ge \xi_{i,j}R^{\min},
\end{aligned} \tag{7}$$

where constraint C1 indicates that each UE can only be associated to one UAV, constraint C2 limits the maximum number of UEs associated with one UAV, where N_0 denotes the maximum number of UEs associated with UAVs. Constraint C3 and C4 are for the UAV deployment, where $x_{\max} = \max\{\bar{x}_i\}$, and $y_{\max} = \max\{\bar{y}_i\}$ denote the maximum values of the positions of users in x and y coordinates, respectively. C5 is the constraint for the available cache capacity of UAVs, C6 and C7 are the constraint for power allocation and UEs minimum required data rate, respectively.

5 UE Clustering, UAV Deployment and Content Caching Strategy

The aim of problem P1 is to minimize overall delay by jointly optimizing UAV deployment, content caching and power allocation strategy. However, the coupling of UAV deployment, content caching, power allocation in (7) makes the optimization problem very challenging to solve directly. To solve this problem, we decompose P1 into tractable sub-problems. In particular, under the assumptions on power allocation, we first formulate and solve UAV deployment and content caching problem in this section. In the next section, we tackle power allocation problem.

5.1 Modified K-Means-Based UE Clustering Algorithm

In this subsection, we design a modified K-means clustering algorithm. In particular, we first define the similarity metrics and discuss the algorithm in detail.

Similarity Metrics. In our considered system model, users may have different content requests. To jointly consider the similarity of users in terms of geographical positions and content requests, we define similarity metrics.

Let $\psi_{\mathrm{d},i,\hat{i}}$ and $\psi_{i,\hat{i},f}$ respectively denote the position similarity between UE_i and $\mathrm{UE}_{\hat{i}}$ and the content similarity of UE_i and $\mathrm{UE}_{\hat{i}}$ on content f, which are expressed as follows

$$\begin{aligned} \psi_{\mathrm{d},i,\hat{i}} &= 1 - \frac{d_{i,\hat{i}}}{\max\limits_{i,\hat{i}} \left\{ d_{i,\hat{i}} \right\}}, \\ \psi_{i,\hat{i},f} &= \begin{cases} \zeta, & \text{if } \gamma_{i,f} = \gamma_{\hat{i},f}, \forall f, i \neq \hat{i} \\ 1, & \text{otherwise}, \end{cases} \end{aligned} \tag{8}$$

where $d_{i,\hat{i}}$ is the distance between UE_i and $\mathrm{UE}_{\hat{i}}$. Users may have similar content requests. To describe the similarity of users on contents, we define content similarity between users. Where ζ is a constant defined to indicate the level of importance of the content similarity, we set $\zeta > 1$.

Accordingly, we define the similarity matrix, $\psi = [\psi_{i,\hat{i}}, \forall i \neq \hat{i}]$, where $\psi_{i,\hat{i}}$ denotes the similarity between UE_i and $\mathrm{UE}_{\hat{i}}$, which is computed as

$$\psi_{i,\hat{i}} = \frac{1}{F}\psi_{i,\hat{i},\mathrm{d}} \sum_{f=1}^{F} \psi_{i,\hat{i},f}. \tag{9}$$

Steps of the Proposed Algorithm. Based on the similarity metric between UEs expressed in (9), we define identifier matrix and select the UE with the highest neighborhood degree as a cluster head to initialize clustering. Then, neighboring users are added to the candidate cluster. Subsequently, distance and cluster size constraints are examined. This process continues until all the remaining users are checked.

The steps of the proposed user clustering algorithm is summarized as follows.

a) Initialization: Let Ω_j, Ω_j^{c} and Ω^{u} denote the set of UEs in the j-th cluster, the j-th candidate cluster and the set of unclustered UEs, respectively. N_j denotes the number of UEs in Ω_j, $d^{\max}$ denotes the maximum radius of the cluster, N_0 stand for the maximum limit of each cluster. We set $\Omega_j = \Omega_j^{\mathrm{c}} = \Omega^{\mathrm{u}} = \emptyset, 1 \leq j \leq J$, $\Omega^{\mathrm{u}} = \Omega^{\mathrm{u}} \cup \{\mathrm{UE}_i, \forall 1 \leq i \leq I\}$, and $j = 1$, where $\emptyset$ denotes the empty set. Let F_i denote the clustering flag of UE_i, we set $F_i = 0$, $1 \leq i \leq I$.

b) Compute similarity metric and identifier index: Compute $\psi_{i,\hat{i}}$ based on (8)-(9), and denote $a_{i,\hat{i}}$ as the identifier index between UE_i and $\mathrm{UE}_{\hat{i}}$. If $\psi_{i,\hat{i}} \geq \psi^{\mathrm{th}}$, we set $a_{i,\hat{i}} = 1$, otherwise, $a_{i,\hat{i}} = 0$, $\forall i \neq \hat{i}$, where ψ^{th} denotes the threshold for similarity.

c) Select the UE with the highest neighborhood degree: Let μ_i denote the neighborhood degree of UE_i, we define μ_i as $\mu_i = \sum_{\hat{i}=1}^{I} a_{i,\hat{i}}, i \neq \hat{i}$. If $\mu_{i^*} = \arg\max_i \{\mu_i\}$, we select UE_{i^*} as the initial user to start clustering and update user sets, i.e., put we put UE_{i^*} in Ω_j^{c} and remove UE_{i^*} from Ω^{u}, i.e., $\Omega_j^{\mathrm{c}} = \Omega_j^{\mathrm{c}} \cup \{\mathrm{UE}_{i^*}\}$, and $\Omega^{\mathrm{u}} = \Omega^{\mathrm{u}} / \{\mathrm{UE}_{i^*}\}$. If $\mu_{i^*} = 1$, jump to Step f).

d) Form candidate cluster Ω_j^{c}: If $\mu_{i^*} \neq 1$, update the candidate cluster Ω_j^{c}. Specifically, randomly select one UE, say $\mathrm{UE}_{\hat{i}}$, if $a_{i^*,\hat{i}} = 1$ and $F_{\hat{i}} = 0$, we check the maximum distance constraint. If $d_{i^*,\hat{i}} \leq d^{\max}$, we put $\mathrm{UE}_{\hat{i}}$ in Ω_j^{c}. i.e., $\Omega_j^{\mathrm{c}} = \Omega_j^{\mathrm{c}} \cup \{\mathrm{UE}_{\hat{i}}\}$, and $\Omega^{\mathrm{u}} = \Omega^{\mathrm{u}} / \{\mathrm{UE}_{\hat{i}}\}$. We set $F_{\hat{i}} = 1$ and $a_{i^*,\hat{i}} = 0$. Repeat this step, until all the remaining UEs are checked.

e) Check the cluster size condition: Compute the size of Ω_j^{c}, i.e., $N_j = \left|\Omega_j^{\mathrm{c}}\right|$. If $N_j \leq N_0$, we set $\Omega_j = \Omega_j^{\mathrm{c}}$, jump to Step f). If $N_j > N_0$, remove the farthest UE from the candidate cluster. That is, if $\hat{i} = \arg\max_{\hat{i}} \left\{d_{i^*,\hat{i}}\right\}$, remove $\mathrm{UE}_{\hat{i}}$ from the candidate cluster, i.e., $\Omega_j^{\mathrm{c}} = \Omega_j^{\mathrm{c}} / \{\mathrm{UE}_{\hat{i}}\}$ and $\Omega^{\mathrm{u}} = \Omega^{\mathrm{u}} \cup \{\mathrm{UE}_{\hat{i}}\}$, we set $F_i = 0$. Repeat this process, until the cluster size condition holds. We set $\Omega_j = \Omega_j^{\mathrm{c}}$.

f) Check algorithm termination: If $\Omega^{\mathrm{u}} = \emptyset$, the algorithm terminates, otherwise, we set $j = j + 1$, return to Step c).

5.2 UAV Deployment Subproblem Formulation and Solution

In this subsection, based on user clustering strategy, we design UAV deployment strategy for individual clusters. Under the assumption that the data transmission in different clusters can be considered independently, we may design the UAV deployment strategy for various clusters individually. Hence, the UAV deployment problem in the system is reduced to the deployment problem for one specific cluster. For convenience, we consider cluster j and design the corresponding UAV deployment strategy, where UAV_j is deployed above cluster j.

Since power allocation strategy may affect the transmission performance between UAVs and UEs, resulting the difficulty in designing UAV deployment. For simplicity, we first apply equal power allocation strategy for cluster j. Specifically, we divide the transmit power equally, i.e., $\alpha^*_{i,j} = 1/N_j, \forall \mathrm{UE}_i \in \Omega_j$.

$$\bar{D}_j = \sum_{i=1}^{N_j} \xi_{i,j} \frac{\sum_{f=1}^{F} \gamma_{i,f}\eta_f}{B\log_2\left(1+\frac{P_j c}{N_j 4\pi f_0(\|\mathbf{q}_j-\mathbf{n}_i\|^2+z^2)\sigma^2}\right)}. \tag{10}$$

UAV deployment subproblem for cluster j can be formulated as

$$\begin{aligned} \mathrm{P2}: \min_{\mathbf{q}_j} \ & \bar{D}_j \\ \text{s.t. } & \mathrm{C3}-\mathrm{C4} \text{ in } (7). \end{aligned} \tag{11}$$

To solve the problem in (11), we employ the quadratic transformation technique on the objective function in $\bar{D}_j$. Specifically, the fractional term X/Y is transformed to the form of $2\kappa\sqrt{X}-\kappa^2 Y$, where $\kappa = \sqrt{X}/Y$. By defining X, Y and κ respectively as $X = \Upsilon_{i,j}$. We denote, $\Upsilon_{i,j}$, ν_j, and $\chi_{i,j}$ by $\xi_{i,j}\sum_{f=1}^{F}\gamma_{i,f}\eta_f$, $\frac{P_j c}{N_j 4\pi f_{m^*}\sigma^2}$, and $\|\mathbf{q}_j-\mathbf{n}_i\|^2$, respectively. Let $\chi^t_{i,j}$ is the value of $\chi_{i,j}$ at the t-th iteration, the content fetching delay at t-th iteration, $\bar{D}^t_j$ can be given as

$$\bar{D}^t_j = \sum_{i=1}^{N_j} \frac{\Upsilon_{i,j}}{B\log_2\left(1+\frac{\nu_j}{\chi^t_{i,j}+z^2}\right)}. \tag{12}$$

We then transform (12) to the form of $2\kappa\sqrt{X}-\kappa^2 Y$, where $\kappa = \sqrt{X}/Y$, and X and Y are respectively the numerator and denominator of the ratio of polynomial expressions.

As $\Upsilon_{i,j}$ and ν_j are both constants, it can be demonstrated that $\lambda^t_{i,j}$ is concave with respect to $\chi^t_{i,j}$. Therefore, $\bar{D}^t_j$ is also a concave function with respect to $\chi^t_{i,j}$. Due to the fact that any concave function is globally upper bounded by its first-order Taylor expansion [18], we apply the first-order Taylor expansion formula on $\bar{D}^t_j$. Let $\hat{\mathbf{q}}^t_j$ denote the local point of $\mathbf{q}^t_j$ at the t-th iteration, applying the first-order Taylor expansion of $\bar{D}^t_j$ with respect to $\chi^t_{i,j}$, we obtain

$$\bar{D}_j^t \leq \sum_{i=1}^{N_j} 2\lambda_{i,j}^t \sqrt{\Upsilon_{i,j}} + \sum_{i=1}^{N_j} B(\lambda_{i,j}^t)^2 H_{i,j}^t (\|\mathbf{q}_j^t - \mathbf{n}_i\|^2 - \|\hat{\mathbf{q}}_j^t - \mathbf{n}_i\|^2) - \sum_{i=1}^{N_j} (\lambda_{i,j}^t)^2 G_{i,j}^t = \hat{D}_j^t \tag{13}$$

where $\hat{D}_j^t$ denotes the upper bound of $\bar{D}_j^t$, $\lambda_{i,j}^t$, H_i^t and $G_{i,j}^t$ are given by

$$\lambda_{i,j}^t = \frac{\sqrt{\Upsilon_{i,j}}}{B \log_2\left(1 + \frac{\nu_j}{\chi_{i,j}^t + z^2}\right)}, H_{i,j}^t = \frac{\frac{\nu_j}{\|\mathbf{q}_j^t - \mathbf{n}_i\|^2} \log_2(e)}{1 + \frac{\nu_j}{\|\mathbf{q}_j^t - \mathbf{n}_i\|^2}},$$
$$G_{i,j}^t = \log_2\left(1 + \frac{\nu_j}{\|\mathbf{q}_j^t - \mathbf{n}_i\|^2}\right) \tag{14}$$

At the t-th iteration, given local point $\hat{\mathbf{q}}_j^t$, the optimization problem (11) can be reformulated as

$$\begin{aligned} \text{P3}: \min_{\{\mathbf{q}_j^t\}} \ & \hat{D}_j^t \\ \text{s.t. } & \text{C3} - \text{C4 in (7)}. \end{aligned} \tag{15}$$

Problem (15) is a convex optimization problem that can be efficiently solved by standard convex optimization solvers such as CVX. Let $\mathbf{q}_j^*$ represent the obtained deployment position of UAV_j.

5.3 UAV Content Caching Subproblem Formulation and Solution

The UAV deployment strategy $\mathbf{q}_j^*$ is obtained under the assumption that users' requested contents are cached at the UAVs. However, this assumption may not hold due to the limited storage capacity of the UAVs. Therefore, efficiently caching the most requested contents and utilizing the limited storage of UAVs become a crucial problem. Since the caching strategy for different UAVs can be designed independently, for simplicity, we formulate and solve content caching problem for UAV_j in this subsection.

Following similar assumptions on transmit power and subchannel allocation strategy in Subsect. 5.2, we compute the content fetching delay of UEs associated with UAV_j. Let $\check{D}_j$ denote the content fetching delay for cluster j, which can be expressed as

$$\check{D}_j = \sum_{i=1}^{I} \xi_{i,j} \left(\frac{\sum_{f=1}^{F} \gamma_{i,f} \eta_f}{B \log_2(1 + \frac{\nu_j}{\|\mathbf{q}_j^* - \mathbf{n}_i\|^2 + z^2})} + \sum_{f=1}^{F} (1 - \delta_{j,f}) \gamma_{i,f} \eta_f D_j^{\text{s}} \right), \forall \text{UE}_i \in \Omega_j. \tag{16}$$

Then, we formulate the content caching subproblem for UAV_j as follows

$$\begin{aligned} \text{P4}: \min_{\{\delta_{j,f}\}} \ & \check{D}_j \\ \text{s.t. C5 in (7).} \end{aligned} \tag{17}$$

To solve P4, we propose a heuristic algorithm that places popular contents in the UAVs. Specifically, we examine the required fetching delay from the content server to UAV_j for various contents. To this end, we denote $\epsilon_f = \sum_{i=1}^{I} \xi_{i,j} \eta_f D_j^{\text{s}}$ as the overall delay incurred to fetch content f from content server. Then, we sort ϵ_f in a descending order. For convenience, we set $\epsilon_{f_1} \geq \epsilon_{f_2} \geq, \cdots, \geq \epsilon_{f_F}$. In order to reduce the backhaul delay, the highest ranking contents should be cached in UAV_j by fulfilling constraint C5. Specifically, if $\eta_{f_1} \leq \rho_j$, we cache content f_1 in UAV_j and we set $\delta_{j,f_1}^* = 1$, otherwise $\delta_{j,f_1}^* = 0$. We then check whether content f_2 can be cached in UAV_j. The above process repeats until no content can be cached in UAV_j under constraint C5 in (7). We use $\delta_{j,f}^*$ to represent the obtained content fetching strategy.

6 Power Allocation Formulation and Solution

Given user clustering, UAV deployment and content caching strategy, which is obtained from Sect. 5, the optimization problem P1 in (7) is reduced to a joint power allocation subproblem. Let $\widetilde{D}$ denote the content fetching delay based on the obtained strategy. The power allocation subproblem can be formulated as

$$\begin{aligned} \text{P5}: \min_{\{\alpha_{i,j}\}} \ & \widetilde{D} \\ \text{s.t. C6} - \text{C7 in (7).} \end{aligned} \tag{18}$$

To solve problem (18), we assume that optimal subcarrier allocation strategy is obtained based on the assumptions made in Subsect. 5.2. Therefore, given user clustering strategy, UAV deployment and content caching strategy for individual clusters, we design power control strategy for users in cluster j, i.e., $\text{UE}_i \in \Omega_j$.

It can be shown that problem P5 is convex which can be solved by using Lagrange dual method. Hence, the corresponding Lagrange function of the problem P5 can be expressed as

$$
\begin{aligned}
&L\left(\alpha_{i,j^*}, \lambda_1, \Omega_i\right) \\
&= \sum_{i=1}^{I} \sum_{f=1}^{F} \left(\frac{\xi_{i,j} \delta^*_{j,f} \gamma_{i,f} \eta_f}{B \log_2 \left(1 + \frac{\alpha_{i,j^*} P_j |h_{i,j^*}|}{\sigma^2}\right)} \right) \\
&+ \lambda_1 \left(\sum_{i=1}^{I} \xi_{i,j} \alpha_{i,j^*} - 1 \right) + \sum_{i=1}^{I} \xi_{i,j} \Omega_i \left(R^{\min} - R_{i,j^*} \right),
\end{aligned} \tag{19}
$$

where λ_1 and Ω_i are non-negative Lagrange multipliers. Then, (18) can be reformulated as

$$
\begin{aligned}
\text{P6}: \max_{\lambda_1, \{\Omega_i\}} \min_{\{\alpha_{i,j^*}\}} & L\left(\alpha_{i,j^*}, \lambda_1, \Omega_i\right) \\
\text{s.t. } & \lambda_1, \Omega_i \geq 0.
\end{aligned} \tag{20}
$$

For a given set of Lagrange multipliers λ_1 and Ω_i , the allocated power α_{i,j^*} can be obtained by calculating the derivative of $L\left(\alpha_{i,j^*}, \lambda_1, \Omega_i\right)$ with respect to α_{i,j^*} and setting it to zero. We set $\alpha^*_{i,j}$ as the obtained power control strategy.

7 Simulation Results

In this section, we evaluate the performance of the proposed BiLSTM-based user request prediction algorithm, user clustering, UAV deployment, content caching, and power allocation strategies. In Table 1, we present the detailed simulation parameters.

We conduct the simulation using the MovieLens dataset [19]. The dataset comprises movie information, such as movie ID and genres, recorded from January 9, 1995, to September 26, 2018. To accurately simulate users' content requests, we selected the eighteen most viewed movie genres based on their view counts. These genres encompass Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Horror, IMAX, Musical, Mystery, Romance, Science Fiction (Sci-Fi), Thriller, War, and Western.

To simulate user requests, we first pre-process the dataset, which involves cleaning the dataset by removing rows with missing values and unnecessary columns. In this simulation, we consider the view counts for each particular genre as requests received from the UEs. We record the request counts for each genre to obtain the request history information for the UEs, as described in (4). Then, we train our model using the request history of the UEs.

Table 1. Simulation parameters.

Simulation Parameters	Notations	Values
Simulation Area		1000 m × 1000 m
Total number of UEs	I	60
Carrier frequency	f_0	2.4 GHz
Bandwidth	B	4 MHz
Noise power	σ^2	−174 dB
Total number of files	F	18
File size	η_f	[5, 6] Mbits
Request rate	λ_j^{q}	$[9, 14]\, b/s$
Service rate	μ_j^{q}	[15, 20]
Maximum number of UEs in clusters	N_0	10
Content similarity	ζ	{1, 1.6}
Threshold for similarity	ψ^{th}	0.7
Maximum radius of clusters	$d^{\max}$	200 m
UAV hovering altitude	z	100 m
Total power of UAV	P_j	1 W
Cache capacity of UAV	ρ_j	36 Mbits
Minimum data rate	$R_{\min}$	0.1 Mbps
Learning rate	λ	0.1

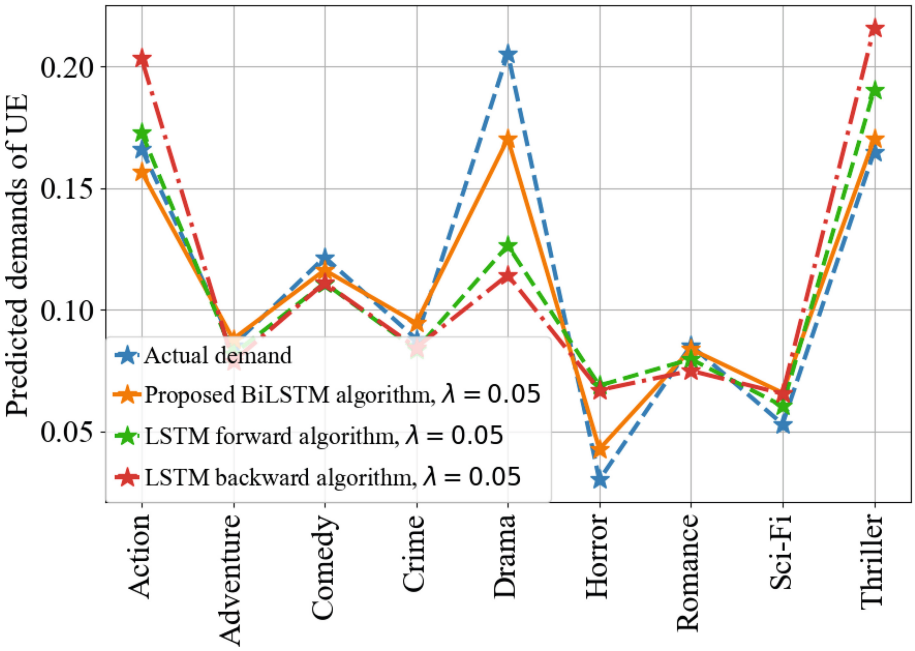

Fig. 2. User content request prediction vs learning rate ($\lambda = 0.05$).

In Figs. 2 and 3, we plot the predicted request of one UE for nine genres. To investigate the prediction accuracy of the proposed BiLSTM-based algorithm versus baseline algorithms, we evaluate the predicted user request versus the actual demand for different learning rates, $\lambda = 0.05$ in Fig. 2, and $\lambda = 0.1$ in Fig. 3. As observed from the figures, the prediction accuracy of the proposed BiLSTM-based algorithm surpasses the baseline algorithms due to its effective learning of UE request history. Thereby indicating the potential advantage that our proposed content caching algorithm achieves compared to the LSTM forward and LSTM backward-based algorithms. The figure demonstrates that for $\lambda = 0.1$ the error difference between the predicted and the actual demands is relatively small compared with $\lambda = 0.05$, which highlights the critical role of the learning rate in enhancing the accuracy of user request prediction.

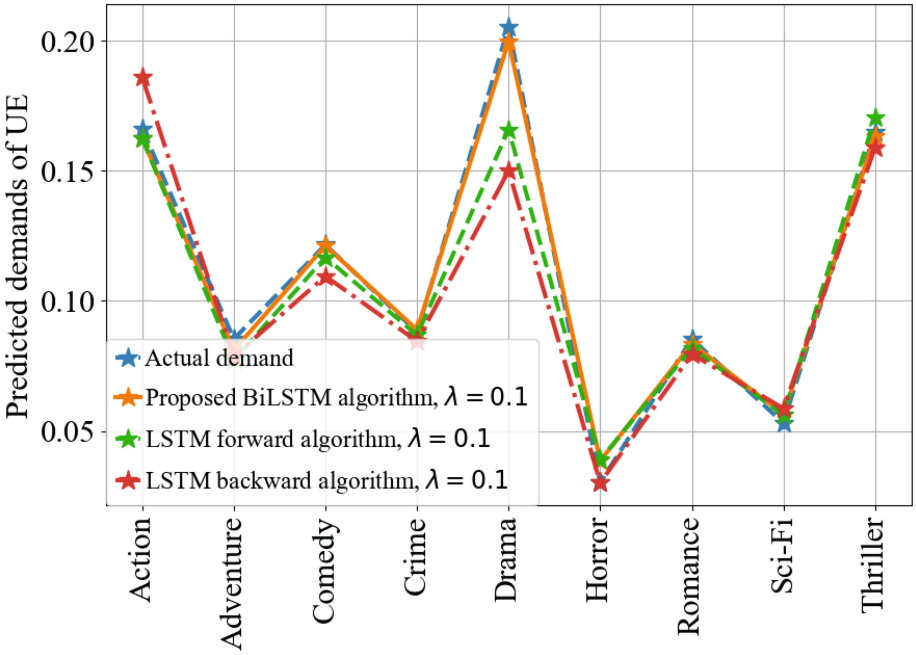

Fig. 3. User content request prediction vs learning rate ($\lambda = 0.1$).

Figure 4, compares UE cluster results with values of $\zeta = 1$ and $\zeta = 1.6$. In the figure, the two numbers in the bracket represent the user ID and request content ID of the user. For instance, Fig. $(40, 12)$ represents that, the content demand of UE_{40} is file $f = 12$. In Fig. 4(a), UE_{40} is clustered with $(7, 14)$ and $(2, 3)$, indicating that UE_2, UE_7, and UE_{40} are grouped based on their geographical location despite differences in their requested content. However, in Fig. 4(b), UE_{40} is clustered with UEs that are both geographically close and have similar content demands, such as $(15, 12)$, $(30, 12)$, $(31, 12)$, and $(51, 12)$. Generally, in

4(b) more UEs with similar content requests are grouped into clusters, emphasizing the influence of user content requests on clustering. The black stars in the figures indicate the positions of the deployed UAVs.

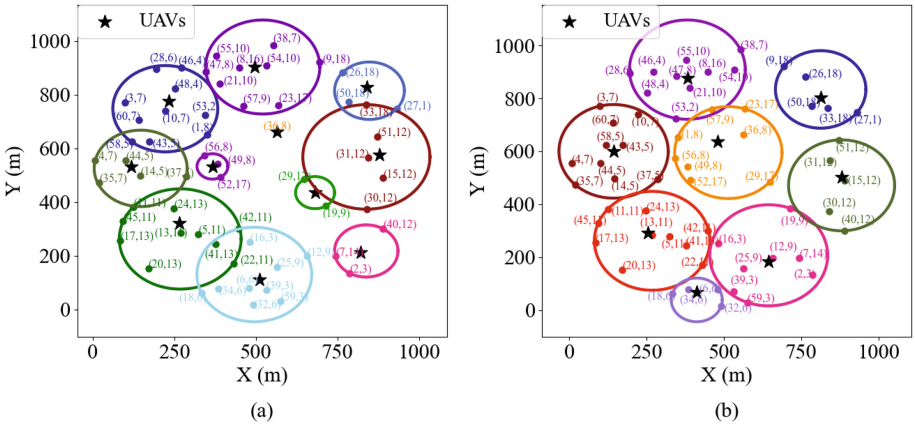

Fig. 4. UE clustering result ((a) $\zeta = 1$ and (b) $\zeta = 1.6$).

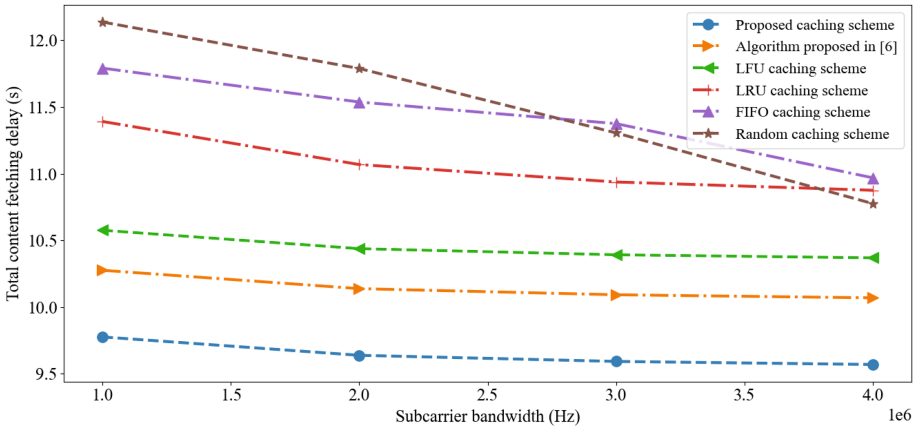

Fig. 5. Total content fetching delay vs subcarrier bandwidth.

Figure 5, plots the total content fetching delay versus bandwidth. The figure presents a comparison of the proposed BiLSTM algorithm against the baseline algorithms, namely, the caching scheme proposed in [6], least frequently used (LFU), least recently used (LRU), first-out (FIFO), and random caching schemes. To evaluate the performance we use the cluster result $\zeta = 1.6$, and utilize the predicted user requests based on the proposed BiLSTM algorithm. It can be observed that the total content fetching delay decreases as bandwidth

increases. Moreover, in comparison to the baseline algorithms, our proposed algorithm achieves the lowest overall content caching delay. This outcome indicates that our proposed algorithms offer the most efficient caching performance. The reason is that our proposed content caching algorithm caches contents that reduce backhaul delay, resulting in an overall reduction of content fetching delay.

In Fig. 6, we evaluate the performance of our proposed algorithms for various UAV cache sizes. The figure plots the total content fetching delay versus the total power of the UAVs. As can be seen from the figure, the total content fetching delay decreases as power increases. Similarly, as the cache size of the UAV increases the overall content fetching delay decreases.

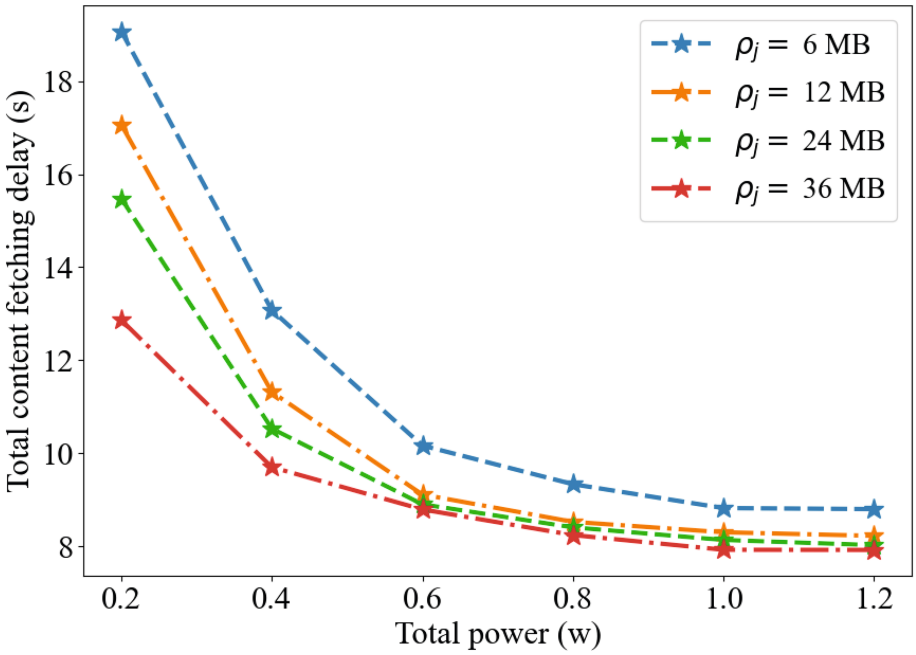

Fig. 6. Total content fetching delay vs UAVs cache sizes.

Figure 7 plots content fetching delay versus cache sizes of the UAVs for various minimum data rates of UEs. As can be seen from the figure, the total content fetching delay decreases as UAV cache size increases. Similarly, as the cache size required minimum data rate of UEs increases the overall content fetching delay of UEs decreases. This is because as the minimum required data rate for each UE increases our proposed power allocation strategy efficiently allocate transmission power for the UEs. This, in turn, lead to reduced overall content fetching delay.

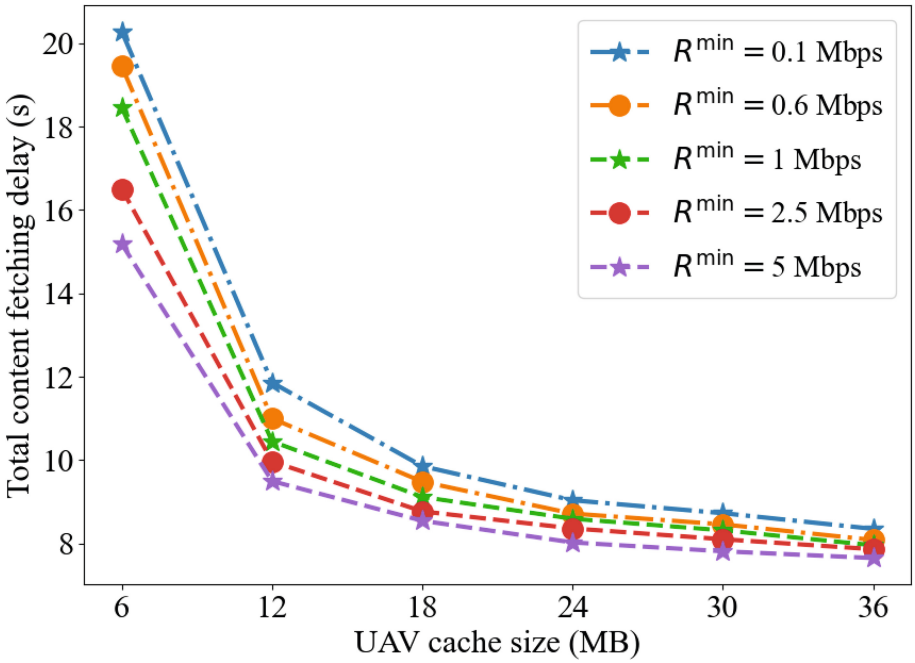

Fig. 7. Total content fetching delay vs UAVs cache sizes.

8 Conclusions

In this paper, we have studied UAV deployment, proactive content caching, and power allocation problems in UAV-enabled networks. Considering the scenario where user content requests are unknown, we developed a BiLSTM-based user request prediction algorithm. Based on the obtained user request prediction and by introducing a content fetching delay function, the joint UE clustering, UAV deployment, content caching, and power allocation problem is formulated as a delay minimization optimization problem, which is solved by applying a heuristic algorithm. Simulation results have shown the effectiveness of the UE clustering, UAV deployment, and content caching strategies in reducing the overall content fetching delay and improving caching performance. In future work, we may extend our current work to a scenario where UAV mobility is considered, and their trajectory is optimized to enhance content-fetching services for UEs. As BiLSTM models rely on fixed-size hidden states, which might not effectively capture variable-length dependencies in sequences, the BiLSTM-based user request prediction can be further studied by considering the request history of users that spans different lengths.

References

1. Khan, M.A., et al.: A survey on mobile edge computing for video streaming: opportunities and challenges. IEEE Access **10**, 120514–120550 (2022). https://doi.org/10.1109/ACCESS.2022.3220694
2. Li, L., Zhao, G., Blum, R.S.: A survey of caching techniques in cellular networks: research issues and challenges in content placement and delivery strategies. IEEE Commun. Surv. Tuts. **20**(3), 1710–1732 (2018). https://doi.org/10.1109/COMST.2018.2820021
3. Li, B., Fei, Z., Zhang, Y.: UAV communications for 5G and beyond: recent advances and future trends. IEEE Internet Things J. **6**(2), 2241–2263 (2019). https://doi.org/10.1109/JIOT.2018.2887086
4. Bhuyan, A.K., Dutta, H., Biswas, S.: Towards a UAV-centric content caching architecture for communication-challenged environments. In: IEEE Global Communications Conference, pp. 468-473, Rio de Janeiro, Brazil, (2022). https://doi.org/10.1109/GLOBECOM48099.2022.10001616
5. Khuwaja, A.A., Zhu, Y., Zheng, G., Chen, Y., Liu, W.: Performance analysis of hybrid UAV networks for probabilistic content caching. IEEE Syst. J. **15**(3), 4013–4024 (2021). https://doi.org/10.1109/JSYST.2020.3013786
6. Kang, M.W., Chung, Y.W.: Content caching based on popularity and priority of content using seq2seq LSTM in ICN. IEEE Access **11**, 16831–16842 (2023). https://doi.org/10.1109/ACCESS.2023.3245803. https://doi.org/10.1109/LWC.2021.3124943
7. Li, D., Zhang, H., Li, T., Ding, H., Yuan, D.: Community detection and attention-weighted federated learning based proactive edge caching for D2D-assisted wireless networks. IEEE Trans. Wirel. Commun. **22**, 7287–7303 (2023). https://doi.org/10.1109/TWC.2023.3249756
8. Wang, E., Dong, Q., Li, Y., Zhang, Y.: Content placement considering the temporal and spatial attributes of content popularity in cache-enabled UAV networks. IEEE Wirel. Commun. **11**(2), 250–253 (2022). https://doi.org/10.1109/LWC.2021.3124943
9. Jiang, B., Yang, J., Xu, H., Song, H., Zheng, G.: Multimedia data throughput maximization in internet-of-things system based on optimization of cache-enabled UAV. IEEE Internet Things J. **6**(2), 3525–3532 (2019). https://doi.org/10.1109/JIOT.2018.2886964
10. Wang, Y., Feng, C., Zhang, T., Liu, Y., Nallanathan, A.: QoE based network deployment and caching placement for cache-enabling UAV networks. In IEEE International Conference on Communications (ICC), pp. 1-6, Dublin, Ireland (2020). https://doi.org/10.1109/ICC40277.2020.9149163
11. Zhang, H., Tang W., Peng, J.: Performance analysis of cooperative caching and transmission diversity in cache-enabled UAV networks. IEEE Trans. Wirel. Commun., 1-1 (2023). https://doi.org/10.1109/TWC.2023.3318110
12. Yin, Y., Liu, M., Gui, G., Gacanin, H., Sari, H., Adachi, F.: Cross-layer resource allocation for UAV-assisted wireless caching networks with NOMA. IEEE Trans. Veh. Technol. **70**(4), 3428–3438 (2021). https://doi.org/10.1109/TVT.2021.3064032
13. Bera, A., Misra, S., Chatterjee, C.: QoE analysis in cache-enabled multi-UAV networks. IEEE Trans. Veh. Technol. **69**(6), 6680–6687 (2020). https://doi.org/10.1109/TVT.2020.2985933

14. Do-Duy, T., Nguyen, L.D., Duong, T.Q., Khosravirad, S.R., Claussen, H.: Joint optimization of real-time deployment and resource allocation for UAV-aided disaster emergency communications. IEEE J. Sel. Areas Commun. **39**(11), 3411–3424 (2021). https://doi.org/10.1109/JSAC.2021.3088662
15. Chen, Y., Zhang, H., Hu, Y.: Optimal power and bandwidth allocation for multiuser video streaming in UAV relay networks. IEEE Trans. Veh. Technol. **69**(6), 6644–6655 (2020). https://doi.org/10.1109/TVT.2020.2985061
16. Kalantari, E., Yanikomeroglu, H., Yongacoglu, A.: Wireless networks with cache-enabled and backhaul-limited aerial base stations. IEEE Trans. Wirel. Commun. **19**(11), 7363–7376 (2020). https://doi.org/10.1109/TWC.2020.3010845
17. Zhou, F., Wang, N., Luo, G., Fan, L., Chen, W.: Edge caching in multi-UAV-enabled radio access networks: 3D modeling and spectral efficiency optimization. IEEE Trans. Signal Inf. Process. Netw. **6**, 329–341 (2020). https://doi.org/10.1109/TSIPN.2020.2986360
18. Duistermaat, J.J., Kolk, J.A.C.: Taylor expansion in several variables. Distributions: Theory and applications. Birkhäuser Boston, Boston (2010)
19. Harper, F.M., Konstan, J.A.: The MovieLens datasets: history and context. ACM Trans. Interact. Intell. Syst **5**, 2160–6455 (2016)