processes

Article

Research on Discrete Artificial Bee Colony Cache Strategy of UAV Edge Network

Yang Hong [1,2], Yuexia Zhang [1,3,*] and Shaoshuai Fan [3]

[1] Key Laboratory of Modern Measurement & Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing 100101, China
[2] Key Laboratory of Information and Communication Systems, Ministry of Information Industry, Beijing Information Science and Technology University, Beijing 100101, China
[3] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: zhangyuexia@bistu.edu.cn

Abstract: Unmanned aerial vehicle edge networks (UENs) can reduce the cache load of the core network and improve system performance to provide users with efficient content services. However, the time-varying characteristics of content popularity in UENs lead to a low accuracy of popularity prediction, and the capacity limitations of wireless channel conditions lead to a lower cache hit rate than the rates of traditional fiber-optic-based cache strategies. Therefore, this paper proposes the discrete artificial bee colony cache strategy of UENs (DABCCSU). First, the information–dynamics–dissemination model of UENs (IDDMU) is established to deduce the coupling relationship between the channel capacity and the service probability in IDDMU. The influence of the service probability change on the content dissemination process is discussed, and the content popularity in UENs is predicted by the state iteration matrix. Then, the discrete artificial bee colony cache (DABCC) optimization algorithm is proposed. The action function of the artificial bee colony is designed as a random action based on the historical cache strategy. The discrete cache strategy is used as an optimization variable, and the popularity prediction result obtained by IDDMU is used to maximize the cache hit rate. DABCC provides the optimal cache strategy for the UENs, and effectively improves the cache hit rate. The simulation result shows that the accuracy of DABCCSU in content popularity prediction is more than 90%, which achieves a good prediction effect. In terms of cache performance, the average cache hit rate of DABCCSU is 91.62%, which is better than the 51.09% of the Least Recently Used (LRU) strategy, 89.27% of the Greedy Algorithm (GA) and 54.26% of Binary Particle Swarm Optimization (BPSO). In addition, the cache hit rate of DABCCSU under different cache capacities is better than that of LRU, GA, and BPSO, showing a relatively stable performance. It shows that DABCCSU can achieve excellent content popularity prediction, and it can also maximize the cache hit rate under limited communication resources and cache resources to provide UENs with the optimal content cache strategy, and provides users with high-quality content services.

Keywords: UAV edge network; popularity prediction; cache strategy; artificial bee colony

Citation: Hong, Y.; Zhang, Y.; Fan, S. Research on Discrete Artificial Bee Colony Cache Strategy of UAV Edge Network. *Processes* **2022**, *10*, 1838. https://doi.org/10.3390/pr10091838

Academic Editor: Xiong Luo

Received: 23 August 2022
Accepted: 9 September 2022
Published: 13 September 2022

1. Introduction

As the important technologies of the next generation communication network, edge networks (ENs) have attracted considerable research interest [1–7]. An edge network caches the content on the edge server and allows users to download interesting content from nearby edge servers. It can effectively cope with the rapid increase in wireless service loads, significantly reducing cache loads and service delays of the core networks, and solves the network congestion problem. Therefore, ENs have become a research hotspot in the field of next generation communication networks.

An unmanned aerial vehicle (UAV) [8,9] has excellent flexibility, mobility, and a unique line-of-sight (LOS) channel from which the UAV edge networks (UENs) [10,11] are derived.

The UENs uses the UAV as a flight relay-assist edge network, which can effectively reduce the cache load of the core network, improve the cache performance of the system, and provide users with efficient content services. The popularity-based edge cache [12–14] is a widely used cache method which uses a large amount of statistical data about content requests to predict content popularity and actively caches high-popularity content from the cloud database in the core network to meet the content needs of users. However, the content popularity changes dynamically with the passage of time. Most popularity-based cache strategies ignore the time variability of content popularity, which makes it difficult to describe the regularity of changes in content popularity. In addition, unlike the edge nodes of the traditional edge cache network that access the cloud database through optical fibers, data transmission between the UENs and the cloud database is realized through wireless communications. This reduced channel capacity seriously restricts the downlink rate between them, limits the content data transmission, and results in a decrease in the cache hit rate. Therefore, cache efficiency of the current popularity-based edge cache strategy is reduced by the limited content transmission between the UENs and cloud database caused by the wireless channel capacity and the low popularity prediction accuracy caused by the time-varying content popularity.

Content popularity prediction aims to improve the cache efficiency of the edge cache network. It needs to accurately capture the dynamic regularity of changes in content popularity. Many scholars have conducted research in related fields [15–20]. Sajad et al. [15] developed a probabilistic dynamics model for content popularity prediction considering the spatial–temporal correlation of content popularity. Kong et al. [16] proposed a popularity prediction method considering the contributions of different dynamic factors and a popularity prediction method based on pattern matching from the micro and macro levels. Fatma et al. [17] proposed a visual social convolutional neural network, which takes the social and visual features of image content into a unified network to predict its popularity. Li et al. [18] studied real data sets from social platforms and proposed a content popularity prediction method based on deep neural networks. Yan et al. [19] solved the content-popularity prediction problem based on the local and global user request states by a machine learning algorithm. Gao et al. [20] proposed the spatial–temporal heterogeneous bassmodel and feature-driven heterogeneous bassmodel to predict the popularity of a single tweet at the early and stable stages. However, studies noted above gave so much attention to content popularity prediction that the coupling relationship between the content popularity and content cache was ignored. Consequently, they have not solved the cache strategy optimization problem.

Currently, many scholars have considered the differences in content popularity and proposed some cache strategies based on content popularity prediction [21–26]. By discovering the correlation between content blocks in information-centric networks, Zhang et al. [21] proposed a block level cache and popularity prediction cache replacement method from the perspective of users. Gao et al. [22] proposed a reinforcement learning model to obtain a cooperative cache strategy based on maximum-distance-separable coding, which captured the time-varying regularity of content popularity. Ji et al. [23] studied the joint content cache and multihop delivery, introduced the distance-sensitive popularity parameter, and proposed a relay-assisted multihop routing algorithm. Liang et al. [24] considered multidimensional features such as historical and future popularity to predict content popularity, proposed a popularity prediction model based on multiheads attention, and then designed a cache strategy according to the prediction results. Chen et al. [25] proposed a popularity prediction framework based on weighted clustering to overcome the sparsity of user requests and considered the similarity of popularity evolution trends to improve cache performance. Liu et al. [26] built a popularity evolution model by analyzing the popularity characteristics of datasets, designed a data-driven popularity prediction method, and proposed a popularity-based eviction and prefetching algorithm to solve the problems of cache content and cache time. However, these studies did not consider the potential impact of wireless channel conditions on the content dissemination process or ignored the time variability of content popularity, which reduces the cache efficiency of the system.

This paper proposes a DABCCSU. This strategy studies the time-variability of content popularity in UEN based on the information dissemination dynamics model [27–29], which can effectively solve the problem of low popularity prediction accuracy caused by the time-varying of content popularity in the UENs. Working against the problem that the cache hit rate of the traditional cache strategy decreases due to the capacity limitation of the wireless channel condition, DABCCSU can maximize the cache hit rate based on popularity prediction of the contents to provide the UENs with the optimal cache strategy, thus improving cache performance. The specific contributions of this paper follow:

(1) An IDDMU is established. Based on this model, the content dissemination process in UEN is analyzed, and the influence of channel capacity on content dissemination results is discussed. Considering the heterogeneity of the dynamic equation, the iterative equation of the state transition is analyzed based on a single user perspective.
(2) A DABCC optimization algorithm is designed. Based on the traditional continuous artificial bee colony algorithm [30,31], the DABCC discretizes the feasible region and redesigns the action function of the artificial bee colony.
(3) A discrete artificial bee colony cache strategy of UEN is proposed. To predict the popularity of the cache content, the strategy obtains the state distribution of UEN users regarding the cache content that is acquired from the user-state iteration matrix of IDDMU. Then, the optimal cache strategy of UAVs is obtained by the DABCC optimization algorithm.

The rest of this paper is organized as follows: In Section 2, the system model is introduced. In Section 3, the cache optimization problem and DABCC algorithm are proposed. In Section 4, the performance of the DABCC algorithm is evaluated, and the simulation results and analysis are given. In Section 5, the conclusion of this paper is presented.

2. System Model

In this section, the channel capacity of UAVs in UEN is analyzed and the IDDMU is established. Then, based on IDDMU, the content dissemination process in UEN is discussed and the iterative matrix for predicting the popularity of cached content is derived.

2.1. UAV Edge Network

The UEN is shown in Figure 1, which includes the cloud database, M UAVs, and N users, where K contents are transmitted among users. The UAV set is represented by $S = \{s_1, s_2, \cdots, s_m, \cdots, s_M\}$, where s_m represents the m-th UAV. The cache capacity set of UAVs is $Ca = \{Ca_1, Ca_2, \cdots Ca_m, \cdots Ca_M\}$, where Ca_m represents the cache capacity of UAV s_m. The user set of the UEN is $U = \{u_1, u_2, \cdots, u_r, \cdots, u_R\}$, where u_r represents the r-th user in the UEN, and all users obey the Poisson distribution. Owing to power limitations, each UAV has a specific coverage. Users within the UAV's coverage are divided into a subset of U, and each user can only communicate with the UAV corresponding to the subset. Therefore, in this paper, the user set U is divided into M subsets $U_1, U_2, \cdots U_m, \cdots U_M$, where $U_m = \{u_{m,1}, u_{m,2}, \cdots, u_{m,n}, \cdots, u_{m,N_m}\}$ represents the user subset within the coverage of s_m, and $u_{m,n}$ represents the n-th user within U_m. It is assumed that the UAV set, and the user set in the UEN are stable, and the content set $F = \{f_1, f_2, \cdots, f_K\}$ is updated in real time. $f_k = \{\theta_k, L_k\}$ represents the k-th content, where $\theta_k \in (0, 1)$ is the reject probability, which represents the average probability that the user is not interested in the content f_k, and L_k (bits) is the size of the content f_k.

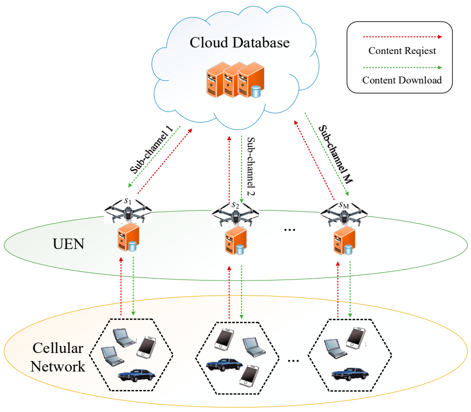

Figure 1. UAV edge network.

In the UEN, the cloud database stores all the contents. Any UAV can function as an edge server in the UEN and cache the high-popularity content from the cloud database according to the popularity prediction. Suppose that the channel capacity between any UAV and the cloud database is C_0. When user $u_{m,n}$ needs to download the content f_k of interest, if s_m has cached the content, $u_{m,n}$ directly downloads it from s_m. Otherwise, $u_{m,n}$ downloads content f_k from the cloud database. This paper mainly studies the cache strategy of the UAV edge server, thus it does not consider the case where users directly obtain content from the cloud database. The UAVs, which work in different frequency bands, update the cache contents from the cloud database in real time by wireless communication. One UAV communicates with the cloud database and all users under its coverage in the same frequency band. Different users covered by the same UAV use time-division duplexing (TDD) technology to communicate with the UAV. The frame structure of UEN is shown in Figure 2, in which the vertical axis shows the frequency band division of different UAVs, and the horizontal axis shows the frame of one UAV. *Frame* $j (j = 0, 1, 2 \cdots)$ is the j-th frame, and the frame length is Δt. Since the cache time of the UAVs and the download time of the users occupy the main part of a frame, the uplink time of the UAV and the user can be flexibly designed, which is not the focus of this paper. This paper mainly considers the downlink part of a frame. One frame is divided into sub-frame 1 and sub-frame 2, each of which is $\Delta t/2$ in length. Sub-frame 1 is divided into multiple time slots and one UAV transmits data to different users in different time slots. During sub-frame 1, users download content from the UAV based on interest, word of mouth, etc. During sub-frame 2, the cloud database predicts the popularity of all content according to the download requests of users in the UEN, designs a cache strategy based on the prediction results, and caches the relevant content to the UAVs. The UAVs in UEN work in different frequency bands and users communicate with UAV in TDD mode. Therefore, this paper ignores the interference between different UAVs and users covered by the same UAV. Assuming that the wireless channel between UAV and user is a LOS channel, the channel capacity is:

$$C_{m,n} = B_m \log\left(1 + \frac{P_m r_{m,n}^{-\xi}}{N_0}\right) \tag{1}$$

where B_m is the working bandwidth of s_m, P_m is the transmission power of s_m, $r_{m,n}$ is the spatial distance between s_m and $u_{m,n}$, N_0 is the Gaussian noise, and ξ is the path loss.

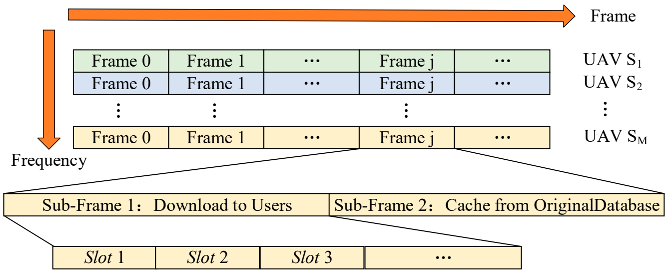

Figure 2. Frame structure of UEN.

2.2. *Information Dissemination Dynamics Model of UEN*

The IDDMU is shown in Figure 3. According to the user's behavior on the content f_k, the four states of the user regarding the content f_k in UEN are defined as follows:

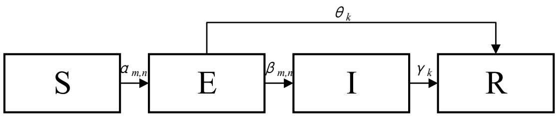

Figure 3. Information dissemination dynamics model of UEN.

S: Never informed on the content f_k. The user may receive recommendations from surrounding users at any time as a potential target.

E: Received the recommendation about content f_k from other users. In this state, there is a certain probability that the user is interested in the content f_k and downloads it from s_m, or is not interested in it and chooses to ignore it.

I: Interested in the content f_k. The user downloaded the content f_k from s_m and recommends it to the surrounding users.

R: Lost interest in content f_k. The user does not recommend content f_k to surrounding users.

$S_{m,k}(t)$, $E_{m,k}(t)$, $I_{m,k}(t)$, and $R_{m,k}(t)$, respectively, represent the number of corresponding state users regarding content f_k covered by s_m at time t, then $S_{m,k}(t) + E_{m,k}(t) + I_{m,k}(t) + R_{m,k}(t) = N_m$. According to the IDDMU, the dissemination dynamic equations of content f_k are obtained as follows:

$$\frac{dS_{m,k}(t)}{dt} = -\sum_{n=1}^{N_m} \alpha_{m,n} S_{m,k}(t) \frac{I_{m,k}(t)}{N_m} \tag{2}$$

$$\frac{dE_{m,k}(t)}{dt} = -\left(\theta_k + \frac{\sum_{n=1}^{N_m} \beta_{m,n}}{N_m}\right) E_{m,k}(t) + \sum_{n=1}^{N_m} \alpha_{m,n} S_{m,k}(t) \frac{I_{m,k}(t)}{N_m} \tag{3}$$

$$\frac{dI_{m,k}(t)}{dt} = -\gamma_k I_{m,k}(t) + \frac{\sum_{n=1}^{N_m} \beta_{m,n}}{N_m} E_{m,k}(t) \tag{4}$$

$$\frac{dR_{m,k}(t)}{dt} = \theta_k E_{m,k}(t) + \gamma_k I_{m,k}(t) \tag{5}$$

The specific definition of the transition probability in Equations (2)–(5) is as follows:

Contact Probability $\alpha_{m,n}$: indicates the probability that $u_{m,n}$ contacts with other users, calculated as:

$$\alpha_{m,n} = \delta \times \varphi_{m,n} \tag{6}$$

In Equation (6), δ is the average probability of successful communication between users and $\varphi_{m,n} = 1 - e^{-\lambda_R \pi R_{m,n}^2}$ is the probability of other users in the vicinity of $u_{m,n}$, where λ_R is the user density of the whole UEN, and $R_{m,n}$ is the communication range of $u_{m,n}$.

Service Probability $\beta_{m,n}$: To ensure that the content f_k can be well presented, the channel capacity between s_m and $u_{m,n}$ must meet certain requirements. Therefore, $\beta_{m,n}$ represents the probability that the channel capacity between the two meets the quality of service (QoS) requirements:

$$\beta_{m,n} = p\left(C_{m,n} \geq C_{m,n}^*\right) \tag{7}$$

where $C_{m,n}^*$ is the minimum channel capacity required for undistorted transmission between s_m and $u_{m,n}$.

According to Equations (1) and (7), the value of $\beta_{m,n}$ depends on the working bandwidth of s_m, the transmission power, and the space distance between s_m and $u_{m,n}$. The specific derivation process of the expression is as follows:

Substituting Equations (1)–(7), leads to:

$$p\left(C_{m,n} \geq C_{m,n}^*\right) = p\left(B_m \log\left(1 + \frac{P_m r_{m,n}^{-\xi}}{N_0}\right) \geq C_{m,n}^*\right) \tag{8}$$

Then, making certain transformations to the right side of the Equation (8) leads to:

$$p\left(C_{m,n} \geq C_{m,n}^*\right) = p\left(r_{m,n} \leq \left(\frac{P_m}{N_0\left(2^{\frac{C_{m,n}^*}{B_m}} - 1\right)}\right)^{\frac{1}{\xi}}\right) \tag{9}$$

All users in the UEN obey the Poisson distribution, thus Equation (9) can be expressed as:

$$p\left(C_{m,n} \geq C_{m,n}^*\right) = \frac{P_m^{\frac{2}{\xi}}}{R_m^2 N_0^{\frac{2}{\xi}}\left(2^{\frac{C_{m,n}^*}{B_m}} - 1\right)^{\frac{2}{\xi}}} \tag{10}$$

where R_m is the coverage range of the UAV s. Thus:

$$\beta_{m,n} = \frac{P_m^{\frac{2}{\xi}}}{R_m^2 N_0^{\frac{2}{\xi}}\left(2^{\frac{C_{m,n}^*}{B_m}} - 1\right)^{\frac{2}{\xi}}} \tag{11}$$

Reject Probability θ_k: indicates the average probability that the user is not interested in the content f_k.

Recovery Probability γ_k: indicates the average probability that users who have downloaded content f_k lose interest in it.

Considering the heterogeneity of ordinary differential equations (Equations (2)–(5)), it is difficult to calculate the specific result of the solution. Therefore, from the perspective of a single user, this paper defines $p_{k,n}^S(t)$, $p_{k,n}^E(t)$, $p_{k,n}^I(t)$, and $p_{k,n}^R(t)$ as the probability that $u_{m,n}$ is in corresponding states with respect to the content f_k at time t, and obtains the iterative equation of the state probability of $u_{m,n}$:

$$p_{k,n}^S(t+1) = \left(1 - p_{k,n}^I \alpha_{m,n}\right)^I p_{k,n}^S(t) \tag{12}$$

$$p_{k,n}^E(t+1) = \left(1 - \left(1 - p_{k,n}^I \alpha_{m,n}\right)^I\right) p_{k,n}^S(t) + \left(1 - \theta_k - \beta_{m,n}\right) p_{k,n}^E(t) \tag{13}$$

$$p_{k,n}^I(t+1) = \beta_{m,n} p_{k,n}^E(t) + \left(1 - \gamma_k\right) p_{k,n}^I(t) \tag{14}$$

$$p_{k,n}^{R}(t+1)=\theta_k p_{k,n}^{E}(t)+\gamma_k p_{k,n}^{I}(t)+p_{k,n}^{R}(t) \tag{15}$$

where l represents the degree of $u_{m,n}$. Converting Equations (12)–(15) into matrix form leads to:

$$\begin{pmatrix} p_{k,n}^{S}(t+1) \\ p_{k,n}^{E}(t+1) \\ p_{k,n}^{I}(t+1) \\ p_{k,n}^{R}(t+1) \end{pmatrix} = \begin{pmatrix} \left(1-p_{k,n}^{I}\alpha_{m,n}\right)^{l} & 0 & 0 & 0 \\ \left(1-\left(1-p_{k,n}^{I}\alpha_{m,n}\right)^{l}\right) & 1-\beta_{m,n}-\theta_k & 0 & 0 \\ 0 & \beta_{m,n} & 1-\gamma_k & 0 \\ 0 & \theta_k & \gamma_k & 1 \end{pmatrix} \begin{pmatrix} p_{k,n}^{S}(t) \\ p_{k,n}^{E}(t) \\ p_{k,n}^{I}(t) \\ p_{k,n}^{R}(t) \end{pmatrix} \tag{16}$$

The state iteration matrix $\mathbf{P}_{k,n}(t)$ is defined as:

$$\mathbf{P}_{k,n}(t) = \begin{pmatrix} \left(1-p_{k,n}^{I}\alpha_{m,n}\right)^{l} & 0 & 0 & 0 \\ \left(1-\left(1-p_{k,n}^{I}\alpha_{m,n}\right)^{l}\right) & 1-\beta_{m,n}-\theta_k & 0 & 0 \\ 0 & \beta_{m,n} & 1-\gamma_k & 0 \\ 0 & \theta_k & \gamma_k & 1 \end{pmatrix} \tag{17}$$

According to $\mathbf{P}_{k,n}(t)$, the proportion of $u_{m,n}$ in each state at time $t+1$ can be predicted as:

$$\mathbf{State}_{k,n}(t+1)=\sum_{n=1}^{N_m}\mathbf{P}_{k,n}(t)\mathbf{State}_{k.n}(t) \tag{18}$$

$$\mathbf{State}_{k,n}(t)=\left(p_{k,n}^{S}(t),p_{k,n}^{E}(t),p_{k,n}^{I}(t),p_{k,n}^{R}(t)\right)^{T} \tag{19}$$

Among the above four states, only $u_{m,n}$ in state E may apply to the UAV for downloading content f_k. Therefore, the cache strategy of UEN is obviously affected by $E_{m,k}(t)$ and $\beta_{m,n}$. Consequently, the prediction popularity of content f_k at time t is defined as:

$$D_{m,k}(t)=E_{m,k}(t)\times\frac{\sum_{n=1}^{N_m}\beta_{m,n}}{N_m} \tag{20}$$

3. Discrete Artificial Bee Colony Cache Strategy of UEN

Based on the content popularity analysis above, this section describes the cache strategy of the UEN and proposes a cache optimization problem. Then, aiming at the optimization problem, DABCCSU is proposed.

3.1. Content Cache

Considering that the content popularity differs among users under different UAV coverage, the UAVs have different cache strategy.

Suppose that $a_{k,m}(t)$ represents the cache of the content f_k by s_m at time t. If s_m caches the content f_k, then $a_{m,k}(t)=1$, otherwise $a_{m,k}(t)=0$. In different cases, s_m has the following four processing methods for content f_k:

(1) $a_{m,k}(t)=1$ and $a_{m,k}(t-1)=1$, s_m retains the content f_k.
(2) $a_{m,k}(t)=1$ and $a_{m,k}(t-1)=0$, s_m caches the content f_k from the cloud database.
(3) $a_{m,k}(t)=0$ and $a_{m,k}(t-1)=1$, s_m deletes the content f_k.
(4) $a_{m,k}(t)=0$ and $a_{m,k}(t-1)=0$, s_m does not process the content f_k.

Thus, the cache of all contents by s_m can be defined as a K-dimensional vector $\mathbf{A}_m(t)=[a_{m,1}(t),a_{m,2}(t),\cdots,a_{m,K}(t)]$.

In the ideal situation without considering constraint conditions such as cache capacity, the cache strategy of UAVs should contain all the content with non-zero popularity in the next frame. However, with the growth of content requirement in UEN, the limited cache capacity of UAVs cannot meet the cache capacity demand of the ideal cache strategy. Moreover, the backhaul capacity between UAVs and cloud database also causes UAVs to be unable to cache content at will. Therefore, this paper defines the cache hit rate of s_m as:

$$h_m(t) = \frac{\sum_{k=1}^{K} D_{m,k}(t) a_{m,k}(t)}{\sum_{k=1}^{K} D_{m,k}(t)} \tag{21}$$

As shown in Equation (21),$h_m(t)$ is actually the ratio of the user contentment of the cache strategy of s_m to the user contentment of the ideal cache strategy, where $\sum_{k=1}^{K} D_{m,k}(t) a_{m,k}(t)$ represents the user contentment of the cache strategy of s_m and is a value based on the weighted sum of popularity. Similarly, $\sum_{k=1}^{K} D_{m,k}(t)$ indicates the user contentment in the ideal situation. If and only if $a_{m,k}(t) = sgn(D_k(t))$, then $h_m(t) = 1$, where $sgn(\cdot)$ is the signum function.

3.2. Cache Optimization Problem

The cache strategy needs to meet the user's content requirements to the maximum extent, that is, to maximize the cache hit rate. The total cache hit rate of UAVs is given in Equation (22):

$$H(t) = \sum_{m=1}^{M} h_m(t) \tag{22}$$

The cache capacity of s_m is limited; as a result, it is impossible to cache all contents in the network at will:

$$\sum_{k=1}^{K} a_{m,k} L_k \leq Ca_m, m = 1, 2, \cdots, M \tag{23}$$

In addition, due to the limitation of channel capacity between the cloud database and the UAVs, it is difficult for s_m to achieve a sharp variation in the cache content within one frame. This is shown in Equation (24):

$$\begin{gathered} \sum_{k=1}^{K} a_{m,k}(t)(1 - a_{m,k}(t-1)) L_k \leq C_d \\ m = 1, 2, \cdots, M \end{gathered} \tag{24}$$

where $a_{m,k}(t)(1 - a_{m,k}(t-1))$ processing methods for content f_k, and $C_d = \frac{1}{2} C_0 \Delta t$ represents the upper limit of bit data transmitted between the UAVs and the cloud database.

The cache optimization problem of the entire UEN can be obtained from Equations (22)–(24):

$$\begin{gathered} \max_{\mathbf{A}} H(t) \\ s.t.(23)(24) \end{gathered} \tag{25}$$

where $\mathbf{A}$ is an M $\times$ K matrix, which represents the cache strategy of the entire UEN. Equation (23) represents the cache capacity constraint of the UAV, and Equation (24) represents the constraint of the channel capacity between UAVs and the cloud database.

3.3. Cache Strategy Optimization

The DABCCSU proposed in this paper includes two parts. First, based on the ID-DMU established in this paper, the content popularity prediction in UEN is obtained by the iteration matrix Equation (17), which was discussed in Section 2.2. Second, based on the prediction results, the DABCC algorithm is proposed to manage the cache optimization problem defined in Equation (25), and then the optimal cache scheme of the UEN is obtained.

The cache optimization problem Equation (25) proposed in this paper contains discrete variables. It is a non-convex integer non-linear programming (INLP) problem and also an NP-complete problem. An exact algorithm such as the enumeration algorithm can obtain the optimal solution of the problem, but its complexity is exponential. Heuristic algorithms such as simulated annealing algorithms can easily fall into local optimal solutions. Therefore, based on the traditional artificial bee colony algorithm, this paper proposes a discrete artificial bee colony cache (DABCC) optimization algorithm.

The cache optimization of UAVs is independent and simultaneous without influence on other UAVs. In other words, the cache strategy of one UAV is only constrained by channel conditions and the cache strategy in the previous frame. Based on this, the optimization problem Equation (25) can be decomposed into the cache optimization of a single UAV s_m, as given in Equation (26):

$$\begin{gathered} \max_{\mathbf{A}_m} h_m(t) \\ s.t. \sum_{k=1}^{K} a_{m,k} L_k \le Ca_m \\ \sum_{k=1}^{K} a_{m,k}(t)(1-a_{m,k}(t-1))L_k \le \tfrac{1}{2}C_0 \Delta t \end{gathered} \tag{26}$$

The optimization problem above can be transformed into a profitability function:

$$\begin{aligned} f_m(t) = h_m(t) - \lambda \max\left(\sum_{k=1}^{K} a_{m,k}(t)L_k - Ca_m, 0\right) \\ - \mu \max\left(\sum_{k=1}^{K} a_{m,k}(t)(1-a_{m,k}(t-1))L_k - \tfrac{1}{2}C_0\Delta t, 0\right) \end{aligned} \tag{27}$$

where $\max(\cdot)$ is a comparison function that outputs the larger of the two parameters. λ and μ are regularization coefficients that are generally large numbers to ensure that the profitability of feasible solutions is greater than infeasible solutions, thus helping eliminate infeasible solutions in time.

The DABCC algorithm follows the definition of the traditional artificial bee colony algorithm and regards the feasible solution as the honey source. The total number of artificial bees is N_{Bee}, which is divided into leader bees, follower bees, and scouter bees. Generally, the number of leader bees and follower bees accounts for half, respectively, i.e., $N_{Bee}/2$. In some cases, scouter bees evolve from leader bees and follower bees. Specific definitions follow:

Honey Source: The honey collection coordinate of the artificial bee colony $Hb_j(j = 1, 2, \cdots, N_{bee}/2)$ represents the honey source of the j-th leader bee. The actions of the artificial bee colony are all centered on the honey sources. In fact, the honey sources are a series of K-dimensional vectors that represent the feasible solution of the optimization problem Equation (26). Artificial bees collect honey at the honey sources and constantly explore nearby honey sources. They compare the profitability of different honey sources by the profitability function and update the optimal honey source in real time. At the initial time, i.e., $t = 0$, since there is no reference honey source, the honey source coordinates are randomly generated:

$$a_{m,k}(t=0) = \begin{cases} 0, rand(0,1) < 0.5 \\ 1, otherwise \end{cases} \tag{28}$$

The honey source coordinates generated by Equation (28) are completely random, which increases the convergence time and calculation cost of the algorithm to a certain extent. Therefore, in the non-initial frame, the DABCC algorithm generates the honey source coordinates based on the cache strategy of the previous frame:

$$a_{m,k}(t) = \begin{cases} 1 - a_{m,k}(t-1), k = \hat{k} \\ a_{m,k}(t-1), otherwise \end{cases} \tag{29}$$

where $\hat{k}$ represents an integer randomly extracted from 1 to K. To ensure the difference between the honey source coordinates, this paper randomly extracts two different integers to execute Equation (29). In general, due to the constraint of channel conditions, the Hamming distance between the honey source generated by Equation (29) and the optimal honey source is smaller than that generated by Equation (28), which reduces the convergence time of the DABCC to a certain extent.

Leader Bee: A leader bee occupies a honey source, explores the nearby honey-source coordinates by the action function, and compares it with the best honey-source coordinates in its memory. When a honey source with a higher profitability is found, the leader bee updates memory and shares it with its follower bee. The leader bee will randomly compare the profitability of the honey source with another leader bee in one iteration. The action function of the leader bee is defined as $\varphi\left(Hb_j, Hb_{j'}\right)$, where Hb_j and $Hb_{j'}$ represent the coordinates of the two paired bees; $\varphi(\cdot)$ retains the same components of the two honey sources and sets the different components to 1 in the form of roulette with the probability calculated by Equation (30):

$$Pr(k) = s_{m,k}(t) / \sum_{k=1}^{K} s_{m,k}(t) \tag{30}$$

where $s_{m,k}(t) = (D_{m,k}(t)/L_k)^Q$ and Q is the weight factor of a positive integer. When $Q = 1$, $s_{m,k}(t)$ represent the popularity gain of a unit content cache bit f_k. As Q increases, the content with the higher popularity gain of unit cache bit becomes more easily cached. The action function can effectively avoid the blind movement of the leader bee and makes it easier to move in the direction of high profitability.

Follower Bee: A follower bee follows a leader bee and explores the nearby honey source. When a better honey source is found by a follower bee, the optimal honey source of its paired leader bee is replaced and the two sides exchange roles. The follower bee selects whether to move using the probability shown in Equation (31). If the follower bee chooses to move, one component of its coordinate is randomly extracted and moved around based on Equation (29):

$$Pm_j = \hat{f}_j / \hat{f}_{max}. \tag{31}$$

where $\hat{f}_j$ represents the profitability of j-th leader bee and $\hat{f}_{max}$ represents the largest profitability of all leader bees. This action of follower bees reduces the calculation cost and ensures the exploration in the direction of high profitability.

Scouter Bee: When the optimal honey source of one leader bee and its follower bee do not change after a certain number of iterations, they are transformed into scouter bees to explore the honey source randomly generated by Equation (28) and redistribute to the leader bee and the follower bee to prevent the DABCC algorithm from falling into the local optimal solution.

The iterative process of DABCC algorithm is shown in Algorithm 1:

Algorithm 1 DABCC

Initialization: at initial time $t = 0$, obtain $\alpha_{m,n}$ of each user; obtain θ_k and γ_k of each content in the initial content set F. Calculate $\beta_{m,n}$ according to the channel conditions between s_m and u_m; Set N_{ite}, N_{lim}, N_{Bee}, λ and μ. Generate initial honey source $Hb_1 \sim Hb_{N_{Bee}}$ by Equation (28).
repeat:
  if $t \neq 0$
    Predict the popularity of content according to Equations (17)–(19).
    Detect $\hat{\alpha}_{m,n}$, $\hat{\theta}_k$, $\hat{\gamma}_k$ and $\hat{\beta}_{m,n}$;
    if $\hat{\alpha}_{m,n} \neq \alpha_{m,n}$ or $\hat{\theta}_k \neq \theta_k$ or $\hat{\gamma}_k \neq \gamma_k$ or $\hat{\beta}_{m,n} \neq \beta_{m,n}$
      Update relevant parameters.
    end
    if $\exists \hat{f}_k \notin F$
      add $\hat{f}_k$ to F.
    end
    Obtain $Pr(k)$ of each content by Equation (30).
end
Match the leader bees and follower bees randomly.
for i = 1:N_{ite}
    for j = 1:$N_{Bee}/2$
      Leader bee Hb_j moves according to $\varphi\left(Hb_j, Hb_{j'}\right)$.
      The follower bee explores the near honey source by Equations (29) and (31).
      Update the roles of leader bee and follower bee and the best honey source.
      Record the iterations N_{none} where the profitability has not improved.
      if $N_{none} = N_{lim}$
        Transform the leader bee and its follower bee into scouter bees for movement.
      end
    end
    Update the best profitability $\hat{f}_{max}$ and its honey source coordinate $Hb_{\max}$.
  end
  Output the cache strategy $\mathbf{A}_m(t) = Hb_{\max}$;
  $t = t + 1$;
until cache task finished.

In the initial frame, the DABCC algorithm initializes the parameters $\alpha_{m,n}$, θ_k, and γ_k of each content in the content set F and calculates $\beta_{m,n}$ according to the channel conditions of s_m. It then sets the iteration times of the algorithm N_{ite}, the maximum iteration times of artificial bees N_{lim}, the honey source dimension K, the punishment factor λ, μ, the number of artificial bees N_{Bee}, and the initial honey source $Hb_1 \sim Hb_{N_{Bee}}$. When the current frame is not the initial frame, the DABCC algorithm first predicts the popularity of the contents according to Equations (**??**)–(**??**), and then detects whether each parameter in the network changes to update it. Next, the DABCC algorithm adds the newly generated content to the content set and executes the action function of the artificial bee. Finally, the algorithm loops into the next frame until the whole cache task ends.

4. Results and Discussion

This section describes the simulation and performance evaluation for the proposed DABCCSU. In this paper, a 5 km ×5 km UEN is generated by MATLAB software to simulate the dissemination of 1000 contents, where users obey the Poisson distribution, and the communication range is 1 km. The generation and dissemination of contents happens randomly. θ_k and γ_k of each content are randomly generated between 0 and 1; $\delta = 0.8$ and $\alpha_{m,n}$ can be obtained by Equation (6). The path loss $\xi = 2$ and $\beta_{m,k}$ is fixed at 0.5 by Equation (11) to facilitate comparison. The parameters are shown in Table 1:

Table 1. Simulation parameters.

Parameter	Value
K	1000
θ_k, γ_k	Rand (0, 1)
$\beta_{m,n}$	0.5
$R_{m,n}$	1 km
ξ	2
C_d	1 Gbits
λ, μ	1×10^{12}
Q	1
N_{Bee}	100
N_{ite}	500
N_{lim}	10

4.1. Content Popularity Prediction by DABCCSU

The prediction results of content popularity by DABCCSU are shown in Figure 4, where the abscissa represents time and the ordinate represents popularity. The two curves are the predicted real content popularities. The sub-graphs (a), (b), and (c) correspond to the situation of UEN corresponding to the number of users N = 100, 300 and 500, respectively. In sub-graph (a), due to the small number of users, the dissemination of content shows randomness, that is, the popularity fluctuates in the early stage and there is a certain amount of error between the prediction result and the real popularity such that the average popularity prediction accuracy is 90.94%. The trend of prediction popularity curves in sub-graphs (b) and (c) is more obvious, and the error between the prediction result and the real popularity is less than sub-graph (a). The average popularity prediction accuracies in sub-graphs (b) and (c) are 92.57% and 93.34%, respectively. Compared with the predicted results of DABCCSU and the real popularity, both trends increase rapidly and then decrease to zero, which conforms to the dissemination content regularity in the network. Once the content is generated, it quickly attracts the interest of surrounding users, and the request for content increases significantly. As the content spreads to saturation, users gradually lose interest in the content, and the popularity of the content rapidly drops to zero. The increased number of users weakens the randomness of content dissemination, making the prediction results more consistent with statistical regularity. Therefore, with an increased number of users, the prediction results of DABCCSU are more accurate.

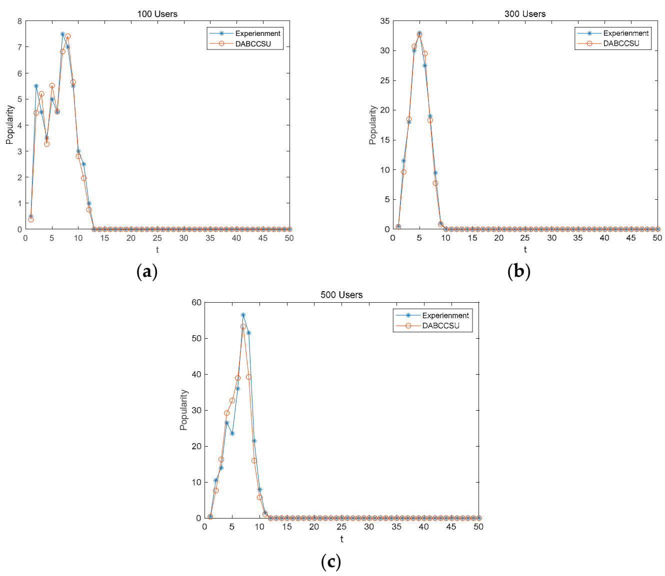

Figure 4. Prediction results of content popularity by DABCCSU: (**a**) 100, (**b**) 300, and (**c**) 500 users.

4.2. DABCC Optimization Algorithm

This paper compares the cache performance of the DABCC optimization algorithm with some common cache algorithms:

(1) Least Recently Used (LRU) Algorithm [32]: Its core idea is that if a content has recent high-frequency requests, it also has a greater probability of being requested currently. When the cache capacity is insufficient, content with a low historical request frequency is preferentially discarded.
(2) Greedy Algorithm (GA): The principle of the GA is to preferentially cache the content with largest $D_{m,k}(t)/L_k$ until the cache capacity or channel load reaches the upper limit.
(3) Binary Particle Swarm Optimization Algorithm (BPSO): BPSO is derived from the particle swarm optimization algorithm, and the value range of its particles is only 0 or1. The number of iterations of the BPSO algorithm in this paper is set to 500 and the number of particles is set to 100. Other parameters are the same as DABCC.

This paper describes the cumulative cache hit rate of the DABCC and reference algorithm in different periods when K = 1000 and the content size is averagely distributed between 0 and 100 Mbits. Figure 5 shows four curves: the cumulative cache hit rates for the DABCC, LRU, GA, and BPSO algorithms, where the abscissa is the time period and the ordinate is the cache hit rate. According to Figure 5, the cumulative cache hit rate of the DABCC is 91.62%, which is much higher than 51.09% for LRU and 54.26% for BPSO and slightly higher than 89.27% for GA. This is because the LRU relies on the historical content request, which makes it difficult to capture the time-variant content popularity. Although the GA considers the time-variant problem of content popularity, it easily falls into the local optimal problem solution. The BPSO converges easily to the local optimal solution, and with the randomness of the search of the algorithm becoming stronger, the local search ability of the BPSO at the later stage of the iteration is weakened. The DABCC effectively avoids this dilemma, resulting in the best cache performance among the four algorithms.

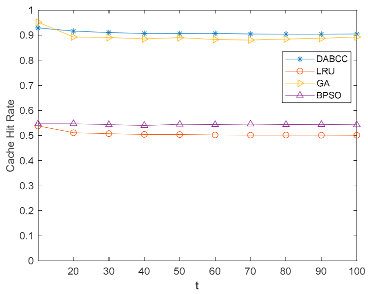

Figure 5. Cumulative cache hit rate.

In addition, this paper discusses the cache hit rate of each algorithm under different cache capacities, as shown in Figure 6. The abscissa represents the cache capacities, which are 0.5, 1, and 2 Gbits. The ordinate is the cache hit rate, and the bar graph represents the average cache hit rate of DABCC, LRU, and GA under different cache capacities. According to Figure 6, the average cache hit rate of the DABCC is higher than that of the other two algorithms regardless of the cache capacity, which conforms to the three algorithms analyzed in this paper. Under any cache capacity, the cache hit rate of the LRU algorithm is only slightly higher than 50%, and cache hit rate of BPSO is about 54%, which are far lower than the other two algorithms. In the case of 0.5 Gbits cache capacity, the DABCC can also achieve an average cache hit rate of 89.83%, which is better than GA's 58.64%. In the case of 2 Gbits cache capacity, the cache hit rate of the DABCC can reach 94.65%, which is slightly better than GA's 93.48%, indicating that the DABCC is more stable in different cache capacities. The LRU has a low cache hit rate because it is difficult to capture the time-variant popularity of the content. The BPSO has a low cache hit rate because of the randomness of particle motion and the lack of local exploration ability in the later stage.

The cache hit rate of the GA is seriously limited by cache capacity. In the case of low cache capacity, the cache hit rate is low owing to the large size content and limited cache capacity. As the cache capacity increases, these restrictions no longer affect the cache efficiency; thus, its cache hit rate can also reach a high level. The DABCC can flexibly design the cache strategy according to the cache capacity, to achieve the maximum cache hit rate under a limited cache capacity. Therefore, the cache hit rate of DABCC is the highest among the three algorithms.

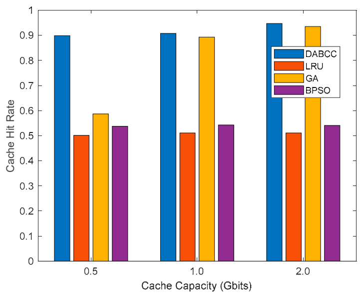

Figure 6. Cache hit rate under different cache capacities.

Figure 7 shows the cache hit rate of DABCC for different iteration times, where the abscissa is the value of N_{ite} and the ordinate is the cache hit rate. The five curves in Figure 7 represent the situation when N_{Bee} = 20, 40, 60, 80, and 100 respectively. It can be understood from Figure 7 that in the case of different N_{Bee}, the cache hit rate of DABCC gradually increases with the increase in the value of N_{ite}. When the value of N_{ite} is large, the cache hit rate tends to be flat. In addition, when N_{Bee} is low, the cache hit rate is significantly lower than when N_{Bee} is high. This phenomenon is in line with the expected results. With the increase in N_{ite}, DABCC gradually approaches the optimal solution, and the cache hit rate rapidly increases. When N_{ite} reaches a certain value, the cache hit rate rises slowly. Because the optimal cache strategy is obtained after a certain number of iterations, the increase in N_{ite} has little impact on the cache hit rate after that. The increase in N_{Bee} improves the efficiency of exploring honey sources in one iteration, resulting in a significantly lower cache hit rate when the N_{Bee} is lower than when N_{Bee} is higher under the same iteration number. Therefore, DABCC proposed in this paper can achieve a cache hit rate of more than 90% with limited N_{ite} and N_{Bee}, which proves DABCC has efficient cache performance.

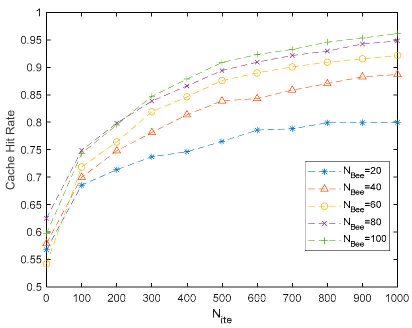

Figure 7. Cache hit rate of DABCC for different iteration times.

5. Conclusions

This paper proposes the discrete artificial bee colony cache strategy of UAV edge network (DABCCSU). The coupling relationship between the popularity and edge cache is derived according to the time-variant characteristics of content popularity in the UEN. In addition, DABCCSU also includes the DABCC optimization algorithm that maximizes the cache hit rate based on the content popularity prediction and provides an optimal cache strategy for the UEN. Simulation results show that the prediction accuracy of DABCCSU is over 90%, and the cache performance has an average cache hit rate of 91.62%, which is better than the LRU and GA strategies. In addition, DABCCSU has a more stable performance under different cache capacities. DABCCSU is expected to be widely used in UAV emergency communication networks or UAV networks in remote areas. In the future, on the basis of this paper, we will continue to study the cache strategy of UENs based on space–air–ground integrated networks, NOMA, intelligent reflect surface, and other technologies.

Author Contributions: Conceptualization, Y.H. and Y.Z.; methodology, Y.H. and Y.Z.; formal analysis, Y.H. and Y.Z.; writing—original draft preparation, Y.H. and Y.Z.; writing—review and editing, Y.H. and Y.Z.; funding acquisition, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by a sub-project of the National Key Research and Development 2020 Plan (2020YFC1511704), Beijing Information Science and Technology University (2020KYNH212, 2021CGZH302), Beijing Science and Technology Project (Z211100004421009), and the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2022-1-16).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Great gratitude to Beijing Information Science and Technology University for providing an excellent experimental environment for the research reported in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fu, Y.; Yu, Q.; Wong, A.K.Y.; Shi, Z.; Wang, H.; Quek, T.Q.S. Exploiting Coding and Recommendation to Improve Cache Efficiency of Reliability-Aware Wireless Edge Caching Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7243–7256. [CrossRef]
2. Qiao, G.; Leng, S.; Maharjan, S.; Zhang, Y.; Ansari, N. Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2020**, *7*, 247–257. [CrossRef]
3. Zhu, X.; Jiang, C.; Kuang, L.; Zhao, Z. Cooperative Multilayer Edge Caching in Integrated Satellite-Terrestrial Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 2924–2937. [CrossRef]
4. Han, Y.; Ai, L.; Wang, R.; Wu, J.; Liu, D.; Ren, H. Cache Placement Optimization in Mobile Edge Computing Networks With Unaware Environment—An Extended Multi-Armed Bandit Approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 8119–8133. [CrossRef]
5. Kwak, J.; Kim, Y.; Le, L.B.; Chong, S. Hybrid Content Caching in 5G Wireless Networks: Cloud Versus Edge Caching. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 3030–3045. [CrossRef]
6. Sun, L.; Zhong, Z.; Qu, Z.; Xiong, N. PerAE: An Effective Personalized AutoEncoder for ECG-Based Biometric in Augmented Reality System. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 2435–2446. [CrossRef] [PubMed]
7. Chen, S.; Yao, Z.; Jiang, X.; Yang, J.; Hanzo, L. Multi-Agent Deep Reinforcement Learning-Based Cooperative Edge Caching for Ultra-Dense Next-Generation Networks. *IEEE Trans. Commun.* **2021**, *69*, 2441–2456. [CrossRef]
8. Ji, J.; Zhu, K.; Niyato, D.; Wang, R. Probabilistic Cache Placement in UAV-Assisted Networks with D2D Connections: Performance Analysis and Trajectory Optimization. *IEEE Trans. Commun.* **2020**, *68*, 6331–6345. [CrossRef]
9. Zhang, M.; I-Hajjar, M.E.; Ng, S.X. Intelligent Caching in UAV-Aided Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 739–752. [CrossRef]
10. Chai, S.; Lau, V.K.N. Multi-UAV Trajectory and Power Optimization for Cached UAV Wireless Networks with Energy and Content Recharging-Demand Driven Deep Learning Approach. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3208–3224. [CrossRef]
11. Luo, J.; Song, J.; Zheng, F.-C.; Gao, L.; Wang, T. User-Centric UAV Deployment and Content Placement in Cache-Enabled Multi-UAV Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 5656–5660. [CrossRef]
12. Bharath, B.N.; Nagananda, K.G.; Gündüz, D.; Poor, H.V. Caching With Time-Varying Popularity Profiles: A Learning-Theoretic Perspective. *IEEE Trans. Commun.* **2018**, *66*, 3837–3847. [CrossRef]
13. Gao, J.; Zhang, S.; Zhao, L.; Shen, X. The Design of Dynamic Probabilistic Caching with Time-Varying Content Popularity. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1672–1684. [CrossRef]
14. Sadeghi, A.; Sheikholeslami, F.; Giannakis, G.B. Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 180–190. [CrossRef]
15. Mehrizi, S.; Chatterjee, S.; Chatzinotas, S.; Ottersten, B. Online Spatiotemporal Popularity Learning via Variational Bayes for Cooperative Caching. *IEEE Trans. Commun.* **2020**, *68*, 7068–7082. [CrossRef]

16. Kong, Q.; Mao, W.; Chen, G.; Zeng, D. Exploring Trends and Patterns of Popularity Stage Evolution in Social Media. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 3817–3827. [CrossRef]
17. Abousaleh, F.S.; Cheng, W.-H.; Yu, N.-H.; Tsao, Y. Multimodal Deep Learning Framework for Image Popularity Prediction on Social Media. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *13*, 679–692. [CrossRef]
18. Li, G.; Liu, Y.; Ribeiro, B.; Ding, H. On New Group Popularity Prediction in Event-Based Social Networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 1239–1250. [CrossRef]
19. Yan, S.; Qi, L.; Zhou, Y.; Peng, M.; Rahman, G.M.S. Joint User Access Mode Selection and Content Popularity Prediction in Non-Orthogonal Multiple Access-Based F-RANs. *IEEE Trans. Commun.* **2020**, *68*, 654–666. [CrossRef]
20. Gao, X.; Zheng, Z.; Chu, Q.; Tang, S.; Chen, G.; Deng, Q. Popularity Prediction for Single Tweet Based on Heterogeneous Bass Model. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 2165–2178. [CrossRef]
21. Zhang, Y.; Tan, X.; Li, W. PPC: Popularity Prediction Caching in ICN. *IEEE Commun. Lett.* **2018**, *22*, 5–8. [CrossRef]
22. Gao, S.; Dong, P.; Pan, Z.; Li, G.Y. Reinforcement Learning Based Cooperative Coded Caching Under Dynamic Popularities in Ultra-Dense Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5442–5456. [CrossRef]
23. Ji, Z.; Wu, S.; Jiang, C.; Wang, W. Popularity-Driven Content Placement and Multi-Hop Delivery for Terrestrial-Satellite Networks. *IEEE Commun. Lett.* **2020**, *24*, 2574–2578. [CrossRef]
24. Liang, J.; Zhu, D.; Liu, H.; Ping, H.; Li, T.; Zhang, H.; Geng, L.; Liu, Y. Multi-Head Attention Based Popularity Prediction Caching in Social Content-Centric Networking with Mobile Edge Computing. *IEEE Commun. Lett.* **2021**, *25*, 508–512. [CrossRef]
25. Chen, Q.; Wang, W.; Yu, F.R.; Tao, M.; Zhang, Z. Content Caching Oriented Popularity Prediction: A Weighted Clustering Approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 623–636. [CrossRef]
26. Chen, B.; Liu, L.; Sun, M.; Ma, H. IoTCache: Toward Data-Driven Network Caching for Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 10064–10076. [CrossRef]
27. Kang, H.; Sun, M.; Yu, Y.; Fu, X.; Bao, B. Spreading Dynamics of an SEIR Model with Delay on Scale-Free Networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 489–496.
28. Sun, L.; Wang, Y.; Qu, Z.; Xiong, N.N. BeatClass: A Sustainable ECG Classification System in IoT-Based eHealth. *IEEE Internet Things J.* **2022**, *9*, 7178–7195. [CrossRef]
29. Guizani, N.; Ghafoor, A. A Network Function Virtualization System for Detecting Malware in Large IoT Based Networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1218–1228. [CrossRef]
30. Karaboga, D. An idea based on Honeybee Swarm for Numerical Optimization. *Tech. Rep.-TR06* **2005**, *200*, 1–10.
31. Yu, Y.; Zheng, J.; Chen, S.; Yang, Z. Moving Target Imaging via Computational Ghost Imaging Combined With Artificial Bee Colony Optimization. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–7. [CrossRef]
32. Kurniawan, F.S.; Yovita, L.V.; Wibowo, T.A. Modified-LRU Algorithm for Caching on Named Data Network. In Proceedings of the 2019 International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 9–10 July 2019.