

Article

NDN-Based Coded Caching Strategy for Satellite Networks

Zhiguo Liu ^{*,†} , Xiaoyong Jin [†], Yifan Li and Luxi Zhang

Communication and Network Key Laboratory, Dalian University, Dalian 116622, China; jxy_dldx@163.com (X.J.); 18290021033@163.com (Y.L.); 15242561513@163.com (L.Z.)

* Correspondence: liuzhiguo@dlu.edu.cn

[†] These authors contributed equally to this work.

Abstract: To solve the transmission correlation issue arising from traditional named data networking (NDN) caching during segmenting contents, where users must obtain data blocks related to the requested file blocks to recover the source data, in this paper, we propose an NDN-based coded caching strategy for low Earth orbit (LEO) satellite networks, using coding operations to remove the transmission correlation of data. To achieve efficient content distribution, the satellite node's coded package is strategically placed with the objectives of minimizing the backhaul link load and content acquisition latency. The optimization problem is solved by using a multi-colony ant colony algorithm, enabling fast content retrieval. We designed a cluster cooperative distribution mechanism to simplify the satellite network's management and reduce the load of satellite links for content distribution when covered by multiple satellite nodes. Finally, we compare the multi-objective coded caching (CCMO) introduced in this paper with the most popular (MP) strategy and the random-based coded caching (CCR) strategy; the average cache hit ratio, backhaul link load, and content acquisition delay are compared and analyzed. The results show that CCMO performs better.

Keywords: named data networking; low Earth orbit satellite networks; coded caching strategy; content distribution; multi-colony ant colony algorithm



Citation: Liu, Z.; Jin, X.; Li, Y.; Zhang, L. NDN-Based Coded Caching Strategy for Satellite Networks. *Electronics* **2023**, *12*, 3756. <https://doi.org/10.3390/electronics12183756>

Academic Editor: Silvia Liberata Ullio

Received: 26 July 2023

Revised: 8 August 2023

Accepted: 9 August 2023

Published: 6 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Low Earth orbit (LEO) satellites can be used as extensions of terrestrial communication networks due to their broad coverage, low latency, and distributed load [1–3]; they also offer advantages in remote areas, mountainous, lake, and maritime networks, emergency communication, data broadcasting, and other scenarios [4,5]. However, due to the limited resources of satellites, satellites face significant energy consumption and communication resource pressures when transferring large amounts of data, which seriously limit improvements in satellite data communication time and quality of service.

Therefore, the idea of exchanging the storage space of satellite nodes with the consumption of the satellite's ground wireless link has been put forward. The cache in the satellite can reduce the communication traffic of the satellite's return link, reduce the communication pressure of the satellite, reduce the service delay of the user, and greatly improve communication efficiency. A new caching scheme, denoted as SatCache, was proposed in [6]; it utilizes the broadcasting characteristics of satellite communication media and creates network user preference profiles to estimate potential interest in a given content. The authors of [7] proposed a double-layer satellite network model for content distribution, which used the satellite as a caching module to cache high-frequency content in the network, and the downlink consumption between the satellite and core network was reduced. At the same time, the cache distribution of the large-scale satellite constellation has also been studied. The authors in [8] proposed a dynamic cache allocation algorithm-based matching game, which considered cache collaboration and cache redundancy between satellites to improve the utilization rate of the cache. The load balancing and allocation of

cache resources in multi-layer satellite networks are investigated and a distributed cache strategy based on the Stackelberg game is proposed in [9].

In recent years, some studies have begun to consider the use of fine-grained coded caching solutions. Coded caching [10] allows files to be divided into coded segments through coding and then caches them into different cache nodes. Users obtain different files from different cache nodes. Coding segments then decode those chunks into the full request file. The authors in [11] proposed a coded caching strategy applied to the satellite's unmanned aerial vehicle (UAV)–vehicle integrated network, which greatly reduced the backhaul traffic between geosynchronous Earth orbit (GEO) satellites and UAVs, saving more transmission energy consumption. In addition, some studies have introduced coded caching into satellite networks to achieve more efficient content distribution. The authors in [12] proposed a heuristic online coded caching scheme based on satellite networks, in which the auxiliary controller searched for opportunities in the aggregation window for linear network coding, according to the content cached in each access satellite. However, this scheme requires the auxiliary controller to go to the ground core end to retrieve the user's required content, code the content, and then distribute the content to LEO satellites. This solution is not very suitable for scenarios where users have large-scale and diverse requests.

With the development of the LEO satellite networks, the future satellite communication networks will develop toward the trend of denseness. With the increase in the number of satellites, on the one hand, terrestrial users can enjoy the coverage of multiple satellite nodes at the same time. On the other hand, terrestrial users can realize the transmission characteristics, such as robustness enhancement and bandwidth aggregation, which are similar to terrestrial networks through multi-point cooperative transmission [13–16].

In this paper, we propose a coded caching scheme of low Earth orbit satellite networks based on named data networking (NDN) [17]. Based on traditional NDN caching for content file segmentation, coding operations are used to remove data transmission correlation, while clustering is carried out based on the geographical location of satellites to simplify the management of satellite networks and reduce the satellite downlink consumption when the user is covered by multiple satellite nodes. Then, according to the different content popularity types, and to minimize the backhaul link load and the link content acquisition delay as the optimization goal, the satellite nodes are placed with different numbers of coded packages, and the optimization problem is solved through the multi-colony ant colony optimization algorithms [18], then an efficient content retrieval and cooperative distribution mechanism is achieved.

The remainder of the paper is organized as follows. Section 2 presents the satellite networks model; we discuss the satellite network model in this paper and the process of clustering satellite nodes. Section 3 presents the coded package naming lookup and cache placement. In Section 4, we discuss the coordinated distribution of the coded packages. In Section 5, the simulation results and discussions are provided and the conclusions are finally drawn in Section 6.

2. Satellite Network Model and Cluster

With the application of satellite network communication and new internet technologies, satellite networks are becoming more dense, and terrestrial users may be covered by multiple satellites within a certain period. As shown in Figure 1, the satellite network architecture of this article is divided into the GEO layer, LEO layer, and terrestrial layer. The control layer of software-defined networking (SDN) [19,20] consists of three GEO satellite constellations, which are responsible for the status collection of LEO satellites and the calculation of cache placement positions. LEO satellites are equipped with NDN routers for in-network caching and content forwarding. The terrestrial layer is composed of Earth stations and content service providers. Since the caching mechanisms in the NDN networks greatly improve the efficiency of content transmission, this paper combines the coded caching strategy and uses multiple satellites to achieve efficient content data distribution.

Assuming that the storage space of the satellite node is C (for the convenience of research, the storage spaces of all satellite nodes are set to be the same in this paper). There are pieces of content in the network, which can be represented as sets $F = \{f_1, f_2, \dots, f_N\}$. The size of each piece of content is M , which can be represented as sets $M = \{m_1, m_2, \dots, m_N\}$. Each content package is divided into k data blocks of equal length, which can be represented as sets $f_1 = \{f_{11}, f_{12}, \dots, f_{1k}\}$. In this paper, the maximum distance separable (MDS) code [21] is used to encode the content data package, and k equal-length data blocks can be coded into any number of coded blocks through the Vandermonde generator matrix, and stored in the satellite nodes. With MDS coding, the satellite node does not need to consider the correlation between content blocks when caching content, as long as the user is guaranteed to receive a sufficient number of coded packages, and then the original data package can be reconstructed.

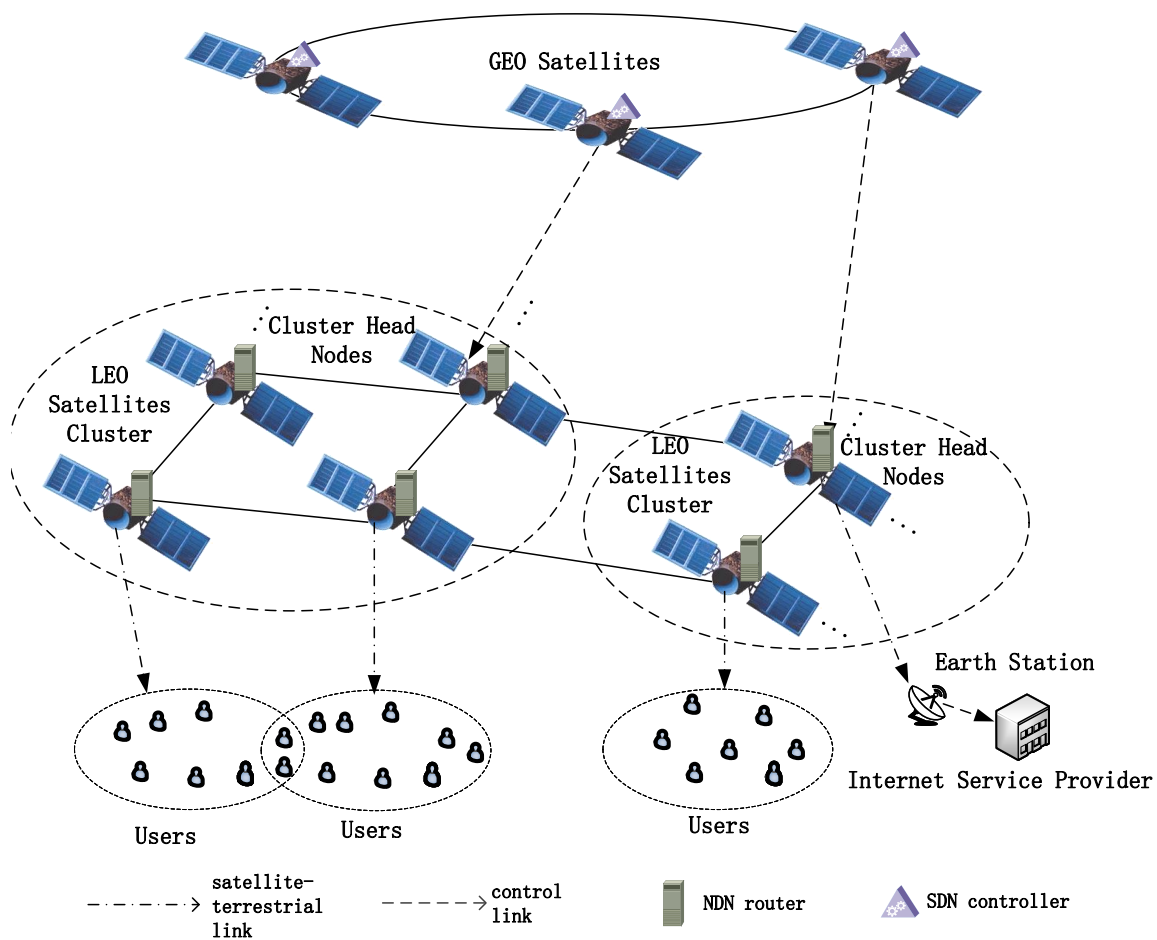


Figure 1. Architecture of the satellite networks.

To simplify the management of the satellite networks and realize the cooperative distribution mechanism of the satellite clusters, this paper divides the clusters into multiple LEO satellite clusters according to the geographic locations of the satellites [22]. The cluster head selection mainly considers the connection degree of the node.

The topology of the satellite networks can be viewed as an undirected graph $G = (V, E)$, where V is the set of nodes in the networks and E is the set of links in the networks. If $\{i, j\} \in E$, there is a bidirectional link between node i and node j , which can communicate with each other.

Due to the predictability and periodicity of the satellite node movement, the satellite operation period can be discretized into N sufficiently small time slices, and it is assumed that the topology of the satellite network does not change in each time slice [7].

The connection degree $degree_i$ of node i in a period can be calculated as follows:

$$degree_i = \frac{1}{N} \sum_{t=0}^N \sum_{j=0, j \neq i}^n a_{ij}^t \quad (1)$$

where N is the number of divided time slices, a_{ij}^t represents the degree of node i in the t time slice. If node i is connected to node j in the t time slice, there is $a_{ij}^t = 1$; otherwise, $a_{ij}^t = 0$.

For satellites S_i and S_j , with the latitude and longitude given, the geocentric angle θ between the satellites is derived by

$$\theta = \arccos[\sin(\varphi_i) * \sin(\varphi_j) + \cos(\varphi_i) * \cos(\varphi_j) * \cos(\lambda_i - \lambda_j)] \quad (2)$$

where φ_i and λ_i represent the latitude and longitude of satellite S_i , and φ_j and λ_j represent the latitude and longitude of satellite S_j .

The distance d_{ij} between node i and node j is derived by:

$$d_{ij} = \sqrt{R_i^2 + R_j^2 - 2R_iR_j \cos \theta} \quad (3)$$

where R_i and R_j are the orbital radii of S_i and S_j , respectively.

The process of clustering satellite nodes is as follows:

- In the initial stage, the node calculates its degree of connection $degree_i$; according to the formula, periodically broadcasts the hello package to exchange information with neighbor nodes, obtains the degree of connection $degree_j$ of neighbor nodes and the distance d_{ij} between nodes, and adds the above information to the node's neighbor list; the list is sorted in the ascending order of the distance between nodes.
- Each node compares the connection degree with its neighbor nodes; if it finds that its connection degree is the largest among the one-hop neighbors, and it is an unclustered node, it will become a cluster head node and broadcast a clustering message.
- All nodes listen to the message. If the node is not the cluster head and receives the clustering message, it sends a joining request to the cluster head node with the highest connection degree. The cluster head node selects nodes to join the cluster from top to bottom according to the neighbor list nodes. The node that successfully joins the cluster broadcasts the joining message. If nodes with higher connection degrees among the neighboring nodes have broadcast the joining message, then the current node is designated as the cluster head node and proceeds to broadcast the clustering message.
- The above process is iterated multiple times until all nodes are classified into the cluster structure. When a node enters a cluster, the head node of the cluster will randomly assign the identity of the node (Node ID) to this node.

After clustering, we can use $W = \{w_1, w_2, \dots, w_u\}$ to represent the cluster of satellite nodes; u is the maximum cluster structure number, $w_u = \{w_{u1}, w_{u2}, \dots, w_{uv}\}$ represents the clustered collection of satellite nodes, and w_{uv} represents the satellite node v of the cluster u .

3. Coded Packages Naming Lookup and Cache Placement

3.1. Naming and Finding Coded Packages

In traditional NDN, each data block has its unique corresponding content name. After network coding is adopted, due to the different naming granularity of interest packages [23], an interest name no longer corresponds to a single data package [24], but to multiple coded packages. The user requests content from the access satellite by sending interest packages. The request arrives at the satellite networks and is forwarded to the cluster head node. The cluster head node checks its cache table. The cache table adopts a secondary directory, as shown in Figure 2. To reduce the number of first-level directory entries in the cache table,

the granularity of the recorded content name is the file level. If the cluster head node finds the corresponding content, it will look for the secondary directory of the cache table. The secondary directory of the cache table is arranged in descending order, according to the number of file-coded blocks, and it also records the Node ID where the coded block is located. The cluster head forwards the request to the corresponding satellite node, and the node retrieves its own content storage (CS) table to find the stored coded package and deliver it.

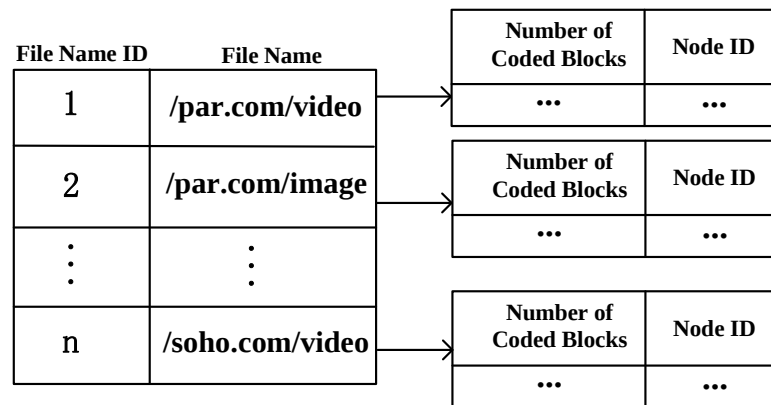


Figure 2. The structure of the cluster head node's cache table.

To reduce the large amount of stored entry data in the satellite node's CS table, this paper uses the naming dictionary tree to store it. Assuming that the content requested by the interest package is /par.com/video, the node will search for the prefix of the local cache entry using the dictionary tree. This search aims to find a matching complete string, extending from the root node of the dictionary tree to any leaf node that aligns with the content concatenation to constitute a content name; at the same time, the child nodes of any node on the concatenation path share the same naming prefix. As shown in Figure 3, when the satellite node retrieves node 4 and completes the matching, the content of the leaf node of node 4 /par.com/video/chunk1 and /par.com/video/chunk2 are the required coded packages.

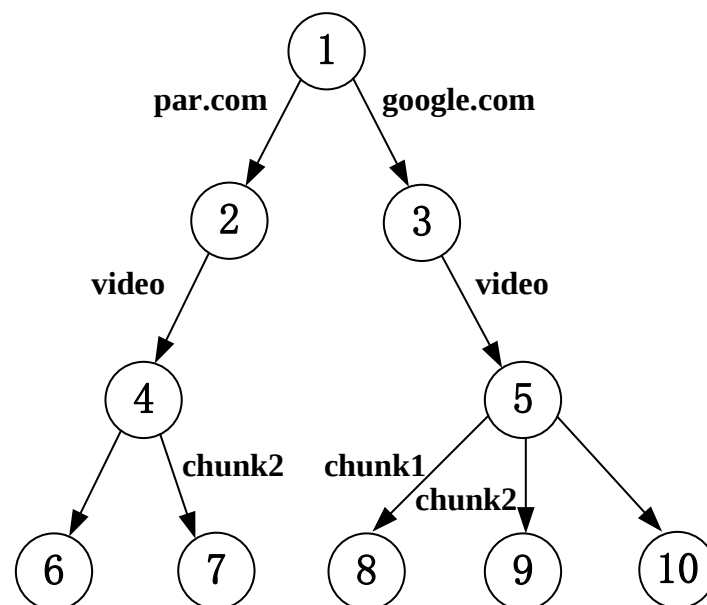


Figure 3. The structure of the naming dictionary tree.

3.2. Quantity Decisions for Coded Packages

In this paper, we consider generating different numbers of coded blocks for content with different popularity types for storage. The more popular content generates more coded blocks, and the less popular content generates fewer coded blocks; the number of cached coded blocks for the content is proportional to its request probability.

$$O_i = p_i \frac{C}{m_{\max}} k \quad (4)$$

and

$$p_i = \frac{1/i^\omega}{\sum_{j=1}^N 1/j^\omega} \quad (5)$$

where O_i represents the number of coded blocks of the i th content, p_i represents the request probability of the content, C is the storage space size of the satellite node, and $m_{\max}$ is the maximum content size.

For a larger p_i , the obtained O_i may be far greater than k , resulting in serious cache redundancy. Therefore, to balance the amount of cached content, the number of coded packages is adjusted based on matching the number of coded packages, and based on the request probability above. The specific algorithm is shown in Algorithm 1.

Algorithm 1 Chart of the coded package number decision algorithm based on request probability matching.

- 1: initialize the number of data packages k , content request probability p_i and number of content N
- 2: **for** $i \leftarrow 1$ to N **do**
- 3: **if** $p_i \frac{C}{m_{\max}} k \leq k$ **then**
- 4: $O_i = p_i \frac{C}{m_{\max}} k$
- 5: **else**
- 6: $O_i = k + \frac{k}{O_i - k} (p_i \frac{C}{m_{\max}} k - k)$
- 7: **end if**
- 8: **end for**

Output: the number of coded packages O_i

If $O_i < k$, it means that the content is not very popular; we then keep the results of the number of coded packages; if $O_i > k$, we map the number $k - O_i$ of coded packages to the interval of $k - 2k$, and round up the final result. Since the content is divided into k data packages, every k -coded package stored by the satellite node is equivalent to storing a copy in the non-coded cache. We cache the same cache benefit, so the upper limit of the number of coded packages is $2k$.

3.3. The Location of the Coded Packages

The placement of the number of satellite-coded packages is expressed as $O_{w_{st,i}} = \{O_{w_{st,1}}, O_{w_{st,2}}, \dots, O_{w_{st,N}}\}$, where $O_{w_{st,i}}$ represents the number of coded packages of the i th file stored on the t th satellite of the s th cluster.

$$\min U = \sum_{i=1}^N p_i \frac{m_i}{k} (k - \min \left\{ \sum_{s=1}^u \sum_{t=1}^v O_{w_{st,i}}, k \right\}) \quad (6)$$

$$\min T = \sum_i p_i R t \left(1 - \operatorname{sgn} \left(\sum_{i=1}^v O_{w_{st,i}} - k \right) \right) + p_i H t \operatorname{sgn} \left(\sum_{i=1}^v O_{w_{st,i}} - k \right) \quad (7)$$

$$f_{\min} = \min[U, T] \quad (8)$$

$$s.t. \begin{cases} \sum_{i=1}^N O_{w_{st,i}} \frac{m_i}{k} \leq C \\ O_{w_{st,i}} = \{0, 1, \dots, k\} \\ \sum_{s=1}^u \sum_{t=1}^v O_{w_{st,i}} \leq O_i \end{cases} \quad (9)$$

where sgn is the step function, when $\sum_{t=1}^v O_{w_{st,i}} - k \geq 0$, the function value is 1, which means that the required coded block can be obtained in a satellite cluster. Assuming that the number of hops in a cluster is H , when $\sum_{t=1}^v O_{w_{st,i}} - k < 0$, the function value is 0, which means that it is necessary to obtain code blocks from other clusters or content servers. Assuming that the number of hops passed at this time is R , $H < R$, t represents the transmission delay of one hop of the inter-satellite link.

The contents are arranged in descending order according to popularity, starting from the most popular content and placing it in sequence. Based on satisfying the cache capacity constraints, we solve the location of each node's coded package to minimize the backhaul link load and content acquisition delay.

3.4. Solving Multi-Objective Optimization Problems Based on the Multi-Colony Ant Colony Algorithm

Since the multiple objectives of the multi-objective optimization problem are mutually constrained and exclusive, each objective cannot reach its optimal value at the same time; generally, only a set of optimal solution sets can be obtained, which is called the Pareto optimal solution [25]. This paper uses a multi-colony ant colony algorithm to solve the optimization problem. The ant colony algorithm is divided into multiple groups according to the number of objective functions, the searches are independent of each other, and the pheromone exchange between the groups is carried out according to certain rules; finally, the Pareto optimal solutions are obtained.

In this algorithm, the solution variable is $O_{w_{st,i}}$, and the value interval of each variable is divided into $\{0, 1, \dots, k\}$ nodes; each $O_{w_{st,i}}$ has $k + 1$ selection nodes. Ant q will choose the path with a high pheromone concentration as the moving direction with a higher probability. The transition probability can be derived as follows:

$$P_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t) = \begin{cases} \frac{[\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)]^\alpha [\eta_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)]^\beta}{\sum_{O_{w_{sr,i}} \in allowed_q} [\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)]^\alpha [\eta_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)]^\beta}, & O_{w_{sr,i}} \in allowed_q \\ 0, & otherwise \end{cases} \quad (10)$$

where $P_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t)$ is the probability that ant q transfers from the node with the number of $O_{w_{st,i}}$ coded blocks under the satellite node w_{st} to the node with the number of $O_{w_{sr,i}}$ coded blocks under the satellite node w_{sr} at time t . $\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)$ is the pheromone concentration on the link at time t . $\eta_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)$ is the heuristic degree of the link at time t , and α and β represent the importance of pheromone and heuristic function factors, respectively; $allowed_q$ is the node set of the number of coded blocks that ant q can still access under the satellite node w_{sr} .

$\eta_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)$ is the expected degree of the ant moving from the $O_{w_{st,i}}$ coded block number node under the satellite node w_{st} to the $O_{w_{sr,i}}$ coded block number node under the satellite node w_{sr} at time t . For the multi-objective optimization model proposed in this paper, the backhaul link load and content acquisition delay are inversely proportional to the expected degree, which can be expressed as the following formula:

$$\eta_{O_{w_{st,i}}, O_{w_{sr,i}}}(t) = \begin{cases} \frac{1}{U_{O_{w_{st,i}}, O_{w_{sr,i}}}} \\ \frac{1}{T_{O_{w_{st,i}}, O_{w_{sr,i}}}} \end{cases} \quad (11)$$

When ant q passes through a path, it will update the pheromone content on the path, so that the pheromone on the path will increase $\Delta\tau_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t)$. When all ants complete an iteration, the pheromone increment on the path will be $\sum_{q=1}^Z \Delta\tau_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t)$; ρ is the pheromone volatile factor, and $0 < \rho < 1$; the local pheromone is updated by the following formula:

$$\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t+1) = (1 - \rho)\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t) + \sum_{q=1}^Z \Delta\tau_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t) \quad (12)$$

Under the backhaul link load target and content acquisition delay target, when the backhaul link load and content acquisition delay of the selected path is smaller, the pheromone concentration on the selected path increases more, inspiring more ants to choose this path. Therefore, the pheromone increase under two targets is set as follows: $U_{O_{w_{st,i}}, O_{w_{sr,i}}}$ and $T_{O_{w_{st,i}}, O_{w_{sr,i}}}$ are the function values of the two targets after the current search is completed, $U_{\min}$ and $T_{\min}$ are the optimal values of the functions of the two targets currently found, and Q is the total amount of pheromones of an ant for one pathfinding.

$$\Delta\tau_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t) = \begin{cases} \frac{Q}{|U_{O_{w_{st,i}}, O_{w_{sr,i}}} - U_{\min}|} & \text{ant } q \text{ passed by } (O_{w_{st,i}}, O_{w_{sr,i}}) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

and

$$\Delta\tau_{(O_{w_{st,i}}, O_{w_{sr,i}})}^q(t) = \begin{cases} \frac{Q}{|T_{O_{w_{st,i}}, O_{w_{sr,i}}} - T_{\min}|} & \text{ant } q \text{ passed by } (O_{w_{st,i}}, O_{w_{sr,i}}) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

After each population completes an independent search, the searched solutions are arranged in ascending order, according to the objective function of the population, and the solution of a certain group is substituted into the objective function of another population from top to bottom. If this solution is better than the worst solution searched for by another population, and this solution is not in the Pareto solution, then we add this solution to the Pareto solution set, and update the global pheromone according to Formula (15), and $0 < e < 1$. If this solution is inferior to the worst solution searched for by another population or the newly searched solution is repeated with a certain Pareto solution in the solution set, we reduce the pheromone on this path and update the pheromone according to Formula (16).

$$\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t+1) = \tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t) + e * Q \quad (15)$$

and

$$\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t+1) = (1 - \rho)\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t) \quad (16)$$

The process of solving the satellite's coded package placement based on the multi-population ant colony algorithm is shown in Algorithm 2.

Algorithm 2 Algorithm for solving the satellite's coded package placement based on the multi-colony ant colony.

```

1: set the number of algorithm iterations  $N_c$ , clear the taboo visited, and initialize parameters  $\alpha$ ,  $\beta$ ,  $\rho$ , and  $Z$ 
2: initialize the pheromone  $\tau_{O_{w_{st,i}}, O_{w_{sr,i}}}(t)$  under the backhaul link load target and the content acquisition, delay the target, respectively
3: place  $Z$  ants of the two populations at the starting node, respectively
4: for  $i \leftarrow 1$  to  $Z$  do
5:   set the taboo table visited index  $s = 1$ 
6:   put the starting code block number node of the  $i$ th ant in the table, visited $i$ 
7:   select the next number of node-coded packages to move forward according to the formula
8:   move the ant to the selected node and insert the node into the table visited $i$ 
9:   if the current node is the destination node then
10:      $s = s + 1$ 
11:   else
12:     select the next node
13:   end if
14: end for
15: locally update the pheromones on each path according to Formula (12)
16: for  $s = 1$  to  $Z$  do
17:   read the full path of  $Z$  nodes in the visited table of the two populations
18:   rank the solutions searched for by the ant colony in ascending order, based on the optimization objectives of the population
19:   if this solution is better than the worst solution searched for by another population and the solution is not in the Pareto solution then
20:     add this solution to the Pareto solution set, and update the pheromone according to Formula (15)
21:   end if
22:   if this solution is inferior to the worst solution found by another population or this solution is already in the Pareto solution then
23:     update the pheromone according to Formula (16)
24:   end if
25: end for
26: set  $N_c = N_c + 1$ 
27: if  $N_c < N_{\max}$  then
28:   clear the table visited, go to step2
29: else
Output: the solution set Pareto
30: end if

```

4. Coordinated Distribution of Coded Packages

To control the distribution of coded packages more precisely, as shown in Figure 4, this paper adds a rank field in the interest package to mark the number of coded blocks required by the user. When the user receives the required number of coded packages, they can decode and restore them to obtain the original data packages. In the process of distributing the coded package, the cluster head node checks the rank field of the interest package and the connection status of each satellite node to control it.

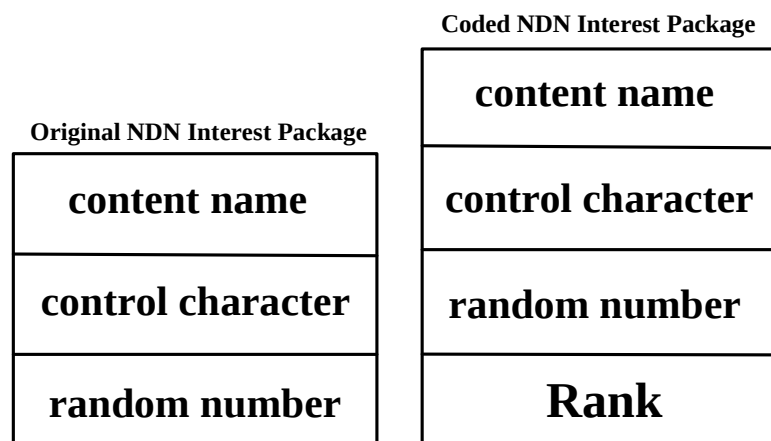


Figure 4. The structure of the original interest package and coded interest package.

If the required coded package is found on the satellite in the cluster and is connected to the user, the cluster head node controls the unified distribution of the coded package of each satellite node. If there is a required coded package on the satellite node but it is not connected to the user, the cluster head node controls each satellite node to forward the coded package to the access satellite node closest to the user and delivers it to the user.

If the cluster head node does not find the required content or the number of coded packages sent by the cluster head node is less than the value of the rank field, i.e., the coded packages in the cluster do not meet the needs of the user. Then, the cluster head node forwards the interest package to the GEO satellite and modifies the value of the rank field, and the GEO satellite retrieves the cache entry table of each cluster except this cluster. If the GEO satellite finds the required content, it forwards the interest package to the corresponding cluster head node; the cluster head node searches the cache table to find the corresponding satellite node, returns all the cached coded packages of the node to the satellite closest to the satellite cluster that issued the original interest package, and then sends them to the user. If the GEO satellite does not retrieve the content requested by the interest package, we forward the request to the ground gateway to obtain the required content from the content server. The distribution process of the coded package is shown in Figure A1.

5. Results

5.1. Simulation Parameters

This article uses the Iridium constellation [26] as the forwarding layer, which contains 66 LEO satellites evenly distributed on 6 orbital planes and uses 3 GEO satellites as controllers to control the entire network in real time. According to [27], this paper sets the content number to 500; the content size in the network is a random number in the range of 1–50 MB; the requests arriving at the node obey the Poisson distribution with $\lambda = 100$ req/s; the parameter ω of Zipf [28] ranges from 0.2 to 1.6; finally, the default value is 0.7.

The value of the parameters in the ant colony algorithm has a great influence on the performance of the algorithm. The authors in [29,30] conducted a large number of experimental analyses on important parameters in the ant colony algorithm and determined the parameters that make the algorithm's convergence speed and solution performance optimal. This article refers to the above references $N_{\max} = 200$, $Z = 100$, $\alpha = 1$, $\beta = 5$, $\rho = 0.7$, $Q = 100$, where $N_{\max}$ represents the maximum number of iterations, Z represents the number of ants in the colony, α represents the pheromone, β represents the importance of heuristic function factors, ρ represents the pheromone volatile factor, Q represents the total amount of pheromone for an ant to find its way at one time.

5.2. Performance Analysis of the Simulation

Based on the above parameters, we compare our coded caching based on multi-objective (CCMO) factors in this paper with the most popular (MP) strategy currently in use and coded caching based on random (CCR). The average cache hit ratio, backhaul link load, and content acquisition delay are compared and analyzed.

Figure 5 shows the distribution diagram of the Pareto solution of the coded caching strategy based on multi-objective optimization, when there are two satellites accessed by users. It can be seen from the figure that the number of Pareto solutions to the cache placement problem in this paper is 14, the distribution is relatively uniform, and each point represents a cache placement strategy that satisfies Pareto.

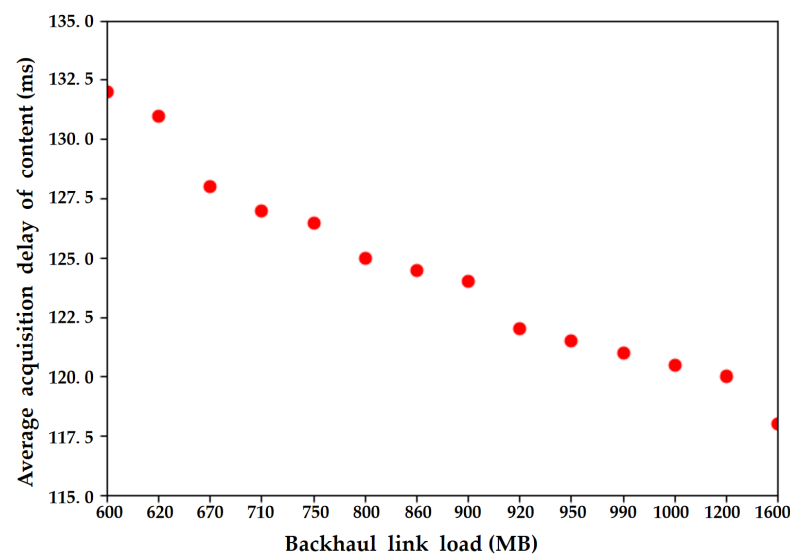


Figure 5. Distribution diagram of Pareto solutions.

Figure 6 shows the relationship between the average acquisition delay of user content and the number of iterations. As the number of iterations continues to increase, the average user content acquisition delay continues to decrease, i.e., the individual fitness in the ant colony algorithm gradually increases, and there is a convergence trend at about 20 iterations; it gradually reaches convergence at about 60 iterations. At the same time, the change data of the average acquisition delay of user content are fitted to better represent the performance of the algorithm.

Figure 7 compares the content acquisition delay of each caching scheme under the number of satellite nodes accessed by different users. From this figure, it can be seen that as the number of users accessing satellite nodes increases, the content acquisition delay of each caching scheme decreases. When the number of satellites accessed by users is 8, compared with MP and CCR, the average acquisition delays of the content of CCMO are reduced by about 25% and 13%. As the number of users accessing satellite nodes increases, the content required by users can be better coordinated and distributed, the user service needs can be met within one time slot as much as possible, and the content acquisition delay of each caching scheme is reduced. However, because the coded cache decouples the transmission correlation between data blocks, only the required coded packages need to be transmitted when transmitting the content required by the user, rather than transmitting all the content, reducing the content acquisition latency compared to the most popular cache placement strategy. Compared with the random-based coded caching strategy, the coded caching strategy based on multi-objective optimization proposed in this paper can select and place popular content more accurately, thereby reducing the content acquisition delay.

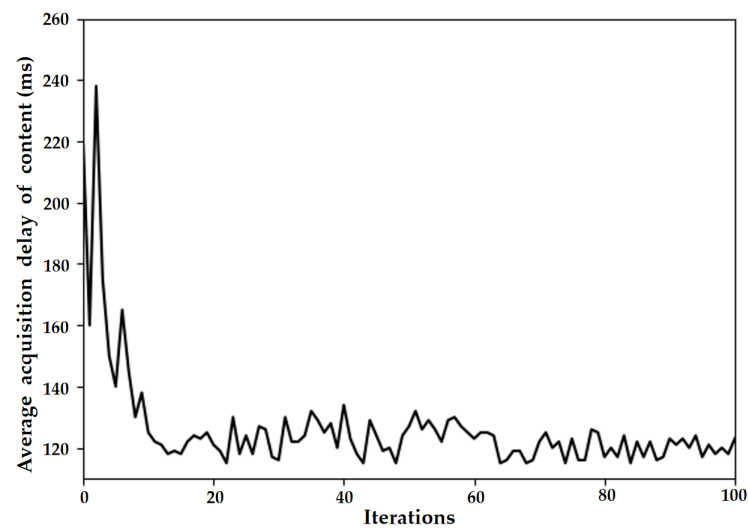


Figure 6. The relationship between the average content acquisition delay and the number of iterations.

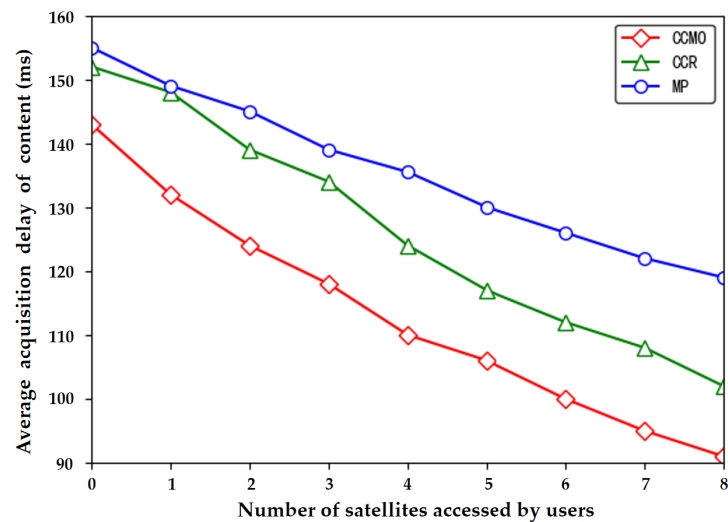


Figure 7. Content acquisition delay diagram under the number of satellite nodes accessed by different users.

Figure 8 compares the backhaul link load of each caching scheme under different user access satellite node numbers. From this figure, it can be seen that as the number of users accessing satellite nodes increases, the backhaul link load of each caching scheme decreases. When the number of satellites accessed by users is 0, the backhaul link load is the same for the three strategies. When the number of satellites accessed by users is 1, compared with MP and CCR, the backhaul link loads of CCMO are reduced by about 46% and 30%. As the number of satellite nodes accessed by users increases, the coordinated delivery of content within a time slot is higher, and coordinated delivery can reduce the load on the backhaul link. Coded caching transforms the smallest content transmission unit into a coded package, leading to a reduced backhaul link load compared to the most popular cache placement strategy. Compared with the random-based coded caching strategy, the code caching strategy based on multi-objective optimization proposed in this paper caches more popular content, and this content will be requested more, so there is a lower backhaul link load.

Figure 9 compares the average cache hit rate of different caching strategies. Compared with MP and CCR, the average cache hit rates of CCMO are improved by about 14% and 30%, respectively. The node cache capacity increases, the content that can be cached

increases, and the cache hit rate naturally increases. At the same time, the average cache hit rate of the CCMO caching strategy proposed in this paper is higher than that of the other two caching strategies. This is because, compared with the non-coded cache, the coded caching strategy in this paper can occupy less cache space to achieve the same cache benefit as the non-coded cache. Compared with the CCR cache strategy, the MP cache strategy can obtain a higher average cache hit rate because the cached content is more popular, and there are relatively more user requests for popular content.

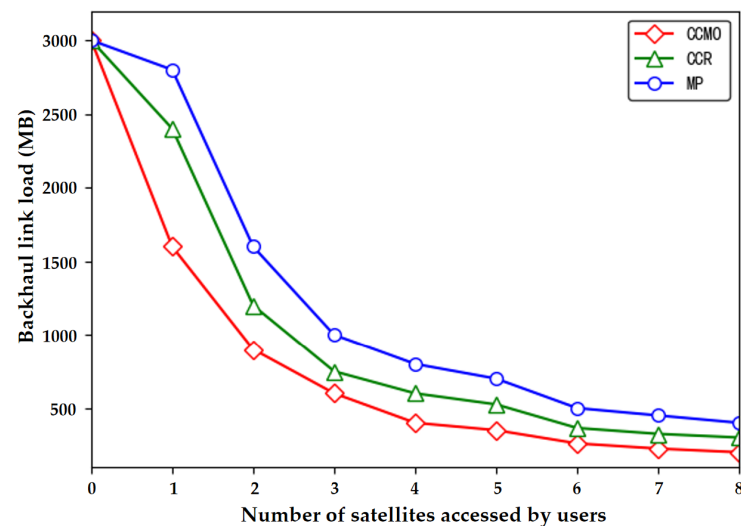


Figure 8. Backhaul link load diagram under the number of satellite nodes accessed by different users.

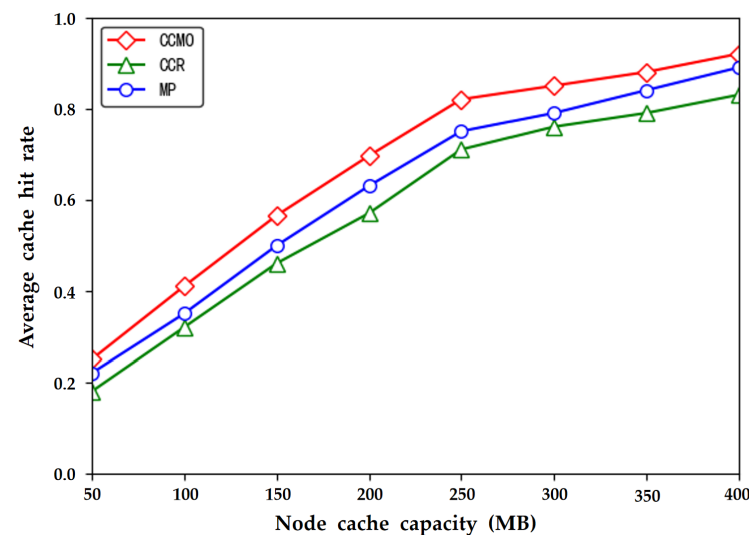


Figure 9. Comparison of the average cache hit rates of different cache strategies.

6. Conclusions

To achieve efficient content distribution in satellite networks, in this paper, we design an NDN-based code cache strategy for low-orbit satellites; we use coding operations to remove the transmission correlation of data, place satellite node-coded packages to minimize backhaul link loads and content acquisition delays, solve the optimization problem through a variety of ant colony algorithms, and realize the rapid retrieval of content. Through the cooperative delivery mechanism of the cluster, the management of the satellite network is simplified, and the user can reduce the load of the satellite link for content delivery when covered by multiple satellite nodes.

In the simulation experiment section, we compare the coded caching based on the multi-objective (CCMO) method proposed in this paper with the most popular (MP) strategy and coded caching based on random (CCR). We conducted a comparative analysis of the indicators, such as the average cache hit rate, backhaul link load, and content acquisition delay. The simulation results show that CCMO performs better in various aspects.

For the coded caching of satellite networks, since the coded caching needs to code the original data blocks, it is necessary to attach the coding coefficient to the header of the generated coded data blocks, so as to facilitate the subsequent linear decoding operation. For the coding overhead of coded caching, further research is needed.

Author Contributions: Conceptualization, Z.L. and X.J.; methodology, Z.L. and X.J.; software, X.J. and Y.L.; validation, X.J. and Y.L.; investigation, X.J., Y.L. and L.Z.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L., X.J., Y.L. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

Appendix A

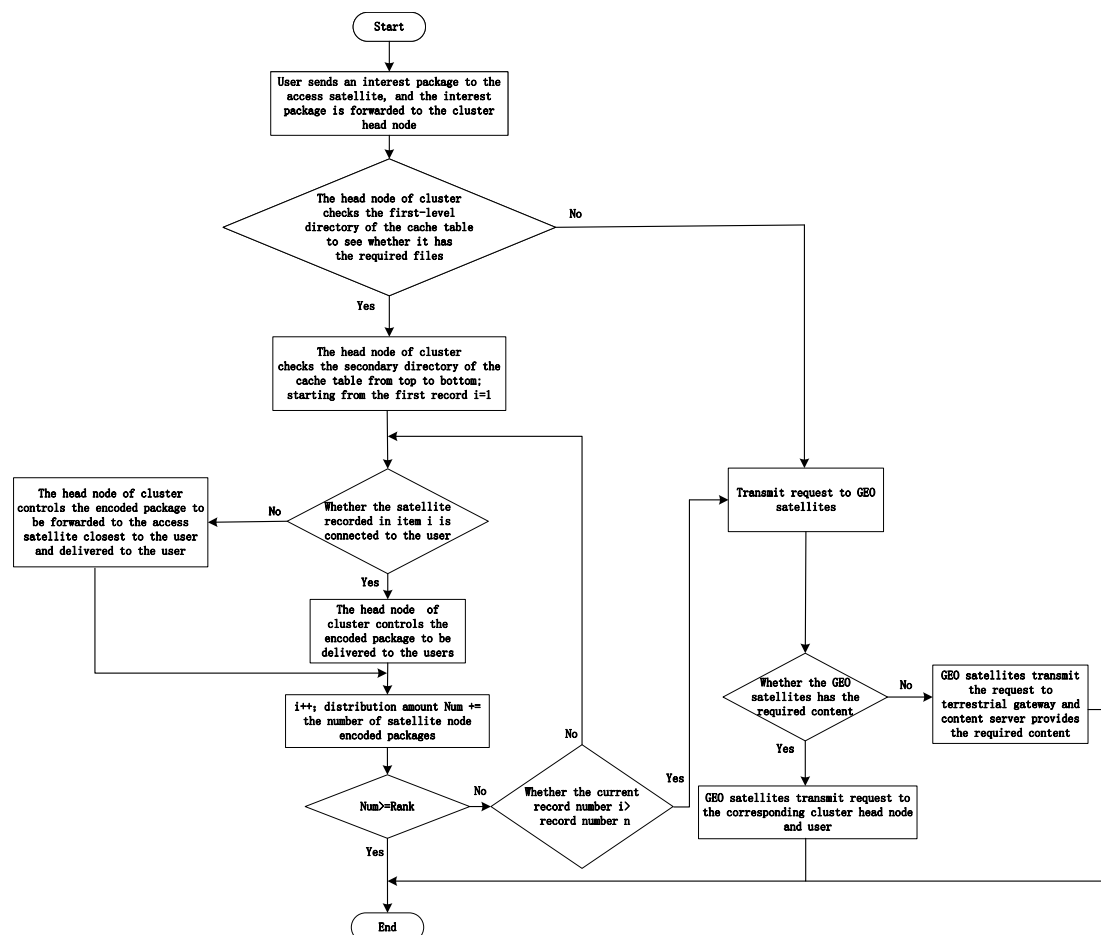


Figure A1. Flow chart of the distributed coded packages.

References

- Jia, X.; Lv, T.; He, F.; Huang, H. Collaborative Data Downloading by Using Inter-Satellite Links in LEO Satellite Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1523–1532. [\[CrossRef\]](#)
- Li, J.; Lu, H.; Xue, K.; Zhang, Y. Temporal Netgrid Model-Based Dynamic Routing in Large-Scale Small Satellite Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6009–6021. [\[CrossRef\]](#)
- Li, J.; Xue, K.; Wei, D.S.L.; Liu, J.; Zhang, Y. Energy Efficiency and Traffic Offloading Optimization in Integrated Satellite/Terrestrial Radio Access Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2367–2381. [\[CrossRef\]](#)
- Chi, L.; Lin, C.; Lin, W.; Liu, Z. Research on development of Space-ground integration information network. In Proceedings of the 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China, 24–26 April 2020; pp. 29–32.
- Zhu, X.; Jiang, C.; Yin, L.; Kuang, L.; Ge, N.; Lu, J. Cooperative Multigroup Multicast Transmission in Integrated Terrestrial-Satellite Networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 981–992. [\[CrossRef\]](#)
- D’oro, S.; Galluccio, L.; Morabito, G.; Palazzo, S. SatCache: A profile-aware caching strategy for information-centric satellite networks. *Trans. Emerg. Telecommun. Technol.* **2014**, *25*, 436–444. [\[CrossRef\]](#)
- Wu, H.; Li, J.; Lu, H.; Hong, P. A Two-Layer Caching Model for Content Delivery Services in Satellite-Terrestrial Networks. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
- Liu, S.; Hu, X.; Wang, Y.; Cui, G.; Wang, W. Distributed Caching Based on Matching Game in LEO Satellite Constellation Networks. *IEEE Commun. Lett.* **2018**, *22*, 300–303. [\[CrossRef\]](#)
- Wang, E.; Li, H.; Zhang, S. Load Balancing Based on Cache Resource Allocation in Satellite Networks. *IEEE Access* **2019**, *7*, 56864–56879. [\[CrossRef\]](#)
- Maddah-alii, M.A.; Niesen, U. Fundamental limits of caching. In Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, 7–12 July 2012; pp. 1077–1081.
- Gu, S.; Sun, X.; Yang, Z.; Huang, T.; Xiang, W.; Yu, K. Energy-Aware Coded Caching Strategy Design with Resource Optimization for Satellite-UAV-Vehicle-Integrated Networks. *IEEE Internet Things J.* **2021**, *9*, 5799–5811. [\[CrossRef\]](#)
- Li, J.; Xue, K.; Liu, J.; Zhang, Y.; Fang, Y. An ICN/SDN-Based Network Architecture and Efficient Content Retrieval for Future Satellite-Terrestrial Integrated Networks. *IEEE Netw.* **2020**, *34*, 188–195. [\[CrossRef\]](#)
- Liu, L.; Garcia, V.; Tian, L.; Pan, Z.; Shi, J. Joint clustering and inter-cell resource allocation for CoMP in ultra dense cellular networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 2560–2564.
- Zhao, G.; Chen, S.; Zhao, L.; Hanzo, L.H. Joint Energy-Spectral-Efficiency Optimization of CoMP and BS Deployment in Dense Large-Scale Cellular Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 4832–4847. [\[CrossRef\]](#)
- Liu, L.; Zhang, S.; Zhang, R. CoMP in the Sky: UAV Placement and Movement Optimization for Multi-User Communications. *IEEE Trans. Commun.* **2018**, *67*, 5645–5658. [\[CrossRef\]](#)
- Tang, J.; Shojaeifard, A.; So, D.K.C.; Wong, K.; Zhao, N. Energy Efficiency Optimization for CoMP-SWIPT Heterogeneous Networks. *IEEE Trans. Commun.* **2018**, *66*, 6368–6383. [\[CrossRef\]](#)
- Paul, R.; Selvan, M.P. A Study On Naming and Caching in Named Data Networking. In Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 11–13 November 2021; pp. 1387–1395.
- Twomey, C.R.; Stützle, T.; Dorigo, M.; Manfrin, M.; Birattari, M. An analysis of communication policies for homogeneous multi-colony ACO algorithms. *Inf. Sci.* **2010**, *180*, 2390–2404. [\[CrossRef\]](#)
- King, D.; Rotsos, C.; Aguado, A.; Georgalas, N.; Lopez, V. The Software Defined Transport Network: Fundamentals, findings and futures. In Proceedings of the 2016 18th International Conference on Transparent Optical Networks (ICTON), Trento, Italy, 10–14 July 2016; pp. 1–4.
- P1930.1/D1 December 2021; IEEE Draft Recommended Practice for Software Defined Networking (SDN) Based Middleware for Control and Management of Wireless Networks. IEEE: New York, NY, USA, 2022; pp. 1–136.
- 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. [\[CrossRef\]](#)
- Qin, J.; Mao, X.; Mcnair, J. Weight based dominating set clustering algorithm for small satellite networks. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 3195–3199.
- Liang, T.; Shi, J.; Wang, Y.; Zhang, B. On the Prefix Granularity Problem in NDN Adaptive Forwarding. *IEEE/ACM Trans. Netw.* **2020**, *29*, 2820–2833. [\[CrossRef\]](#)
- Kim, J.; Ko, M.C.; Shin, M.S.; Kim, J. Scalable Name Lookup for NDN Using Hierarchical Hashing and Patricia Trie. *Appl. Sci.* **2020**, *10*, 1023. [\[CrossRef\]](#)
- Li, M.; Yang, S.; Liu, X. Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 645–665. [\[CrossRef\]](#)
- Fossa, C.; Raines, R.; Gunsch, G.; Temple, M. An overview of the IRIDIUM (R) low earth orbit (LEO) satellite system. In Proceedings of the IEEE 1998 National Aerospace and Electronics Conference. NAECON 1998. Celebrating 50 Years (Cat. No.98CH36185), Dayton, OH, USA, 17 July 1998; pp. 152–159.

27. 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](#)]
28. Hachem, J.; Karamchandani, N.; Moharir, S.; Diggavi, S. Caching with partial matching under Zipf demands. In Proceedings of the 2017 IEEE Information Theory Workshop (ITW), Solstrand, Norway, 1–6 July 2017; pp. 61–65.
29. Wei, X.; Li, Z. Research on Parameters Optimization and Simulation of the Ant Colony Algorithm. In Proceedings of the International Conference on Cyberspace Technology (CCT 2014), Beijing, China, 8–10 November 2014.
30. Shrivastava, K.; Kumar, S. The Effectiveness of Parameter Tuning on Ant Colony Optimization for Solving the Travelling Salesman Problem. In Proceedings of the 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 24–26 November 2018; pp. 78–83.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.