

Article

User Experience-Oriented Content Caching for Low Earth Orbit Satellite-Enabled Mobile Edge Computing Networks

Jianhua He ¹, Youhan Zhao ², Yonghua Ma ¹ and Qiang Wang ^{1,*}

¹ Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China; hejianhua@csu.ac.cn (J.H.); mayonghua@csu.ac.cn (Y.M.)

² State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; zhaoyouhan@bupt.edu.cn

* Correspondence: wq@csu.ac.cn

Abstract: In this paper, we investigate a low Earth orbit (LEO) satellite-enabled mobile edge computing (MEC) network, where multiple cache-enabled LEO satellites are deployed to address heterogeneous content requests from ground users. To evaluate the network's capability in meeting user demands, we adopt the average quality of experience (QoE) of the users as the performance metric, defined based on the effective transmission rate under communication interference. Our analysis reveals that the average QoE is determined by the content caching decisions at the satellites, thereby allowing us to formulate an average QoE maximization problem, subject to practical constraints on the satellite caching capacity. To tackle this NP-hard problem, we design a two-stage content caching algorithm that combines divide-and-conquer and greedy policies for efficient solution. The numerical results validate the effectiveness of the proposed approach. Compared with several benchmark schemes, our algorithm achieves notable improvements in terms of the average QoE while significantly reducing caching costs, particularly under resource-constrained satellite settings.

Keywords: content caching; quality of experience (QoE); mobile edge computing (MEC); satellite-terrestrial integrated networks



Academic Editor: Juan-Carlos Cano

Received: 22 April 2025

Revised: 22 May 2025

Accepted: 9 June 2025

Published: 13 June 2025

Citation: He, J.; Zhao, Y.; Ma, Y.; Wang, Q. User Experience-Oriented Content Caching for Low Earth Orbit Satellite-Enabled Mobile Edge Computing Networks. *Electronics* **2025**, *14*, 2413. <https://doi.org/10.3390/electronics14122413>

Copyright: © 2025 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

Sixth-generation (6G) wireless networks are envisioned to revolutionize global communications by enabling ubiquitous connectivity, ultra-low latency, massive device access, and seamless integration of physical and digital worlds. As a key enabler of 6G systems, integrated ground-air-space networks have emerged as a fundamental component to ensure wide-area, resilient, and continuous service coverage [1]. Among their components, low earth orbit (LEO) satellite communications stand out for their advantages of global coverage, fast deployment, infrastructure independence, and high resilience to disasters [2]. Unlike terrestrial systems, which are constrained by infrastructure availability, and unmanned aerial vehicle (UAV)-based platforms with limited spatial and temporal coverage, LEO satellites enable continuous wide-area services across both populated and remote regions. Current applications of LEO satellite systems include global broadband access (e.g., Starlink by SpaceX) [3], oceanic and aerial navigation [4], remote sensing data support [5], and emergency communications [6]. These functions not only extend the reach of traditional networks but also lay a foundation for robust and intelligent 6G infrastructures on a global scale.

Meanwhile, to alleviate the increasing computational burden on user devices and reduce reliance on distant cloud infrastructures, mobile edge computing (MEC) has gained significant traction as a decentralized computing paradigm [7]. By bringing processing capabilities closer to end users—at the edge of the network—MEC effectively reduces latency, conserves bandwidth, and enhances responsiveness. In recent years, research efforts have converged on optimizing MEC strategies to improve the overall quality of experience (QoE) for users [8–13]. Two central aspects in this domain are computation offloading and service and content caching. The former allows user devices to delegate intensive tasks to nearby edge servers, thereby improving execution efficiency and device energy consumption. The latter focuses on proactively storing popular content, applications, or service data at edge nodes to enable fast and localized data access, which is particularly critical for latency-sensitive applications. For instance, the authors of [9] jointly optimized task scheduling and offloading decisions for time-sensitive applications, aiming to minimize the system's age of information (AoI). In [10], a task migration scheme among MEC servers was proposed to reduce the overall task completion delay. To balance computational load across edge nodes, a graph neural network-based method was introduced in [11]. These approaches generally assume that the edge servers have already cached the required content or services [8,14] and that users can readily establish high-quality communication links with edge nodes. However, this assumption overlooks the fact that caching and content retrieval are equally critical components of edge computing performance, yet they have received comparatively less attention. To address this issue, the authors of [12] investigated joint optimization of service placement and computation offloading in ultra-dense small cell networks, targeting the reduction of service delays and energy consumption. In terms of caching and downloading, the authors of [15] considered caching popular content over unreliable resources as a cost-effective solution to limited edge storage, enhancing both content availability and access rates during peak periods.

With the growing recognition of the potential of LEO satellite networks to support both wide-area communication and distributed caching in MEC networks, increasing efforts have been directed toward incorporating caching mechanisms into LEO-based architectures [16]. In [17], an LEO satellite constellation-enabled service caching system was considered. In this system, the problem of efficiently deploying services on satellite nodes to realize robustness-aware service coverage with constrained resources was investigated. An online service placement algorithm with a theoretical performance guarantee through leveraging Lyapunov optimization and Gibbs sampling was proposed to improve the service coverage. In [18], a cooperative transmission scheme in integrated satellite-terrestrial (IST) networks was considered to cope with the dynamic satellite time-varying topology and propagation links. A low-complexity greedy algorithm was designed to minimize the average file transmission delay by jointly optimizing cache placement and content scheduling based on user requests. In [19], the cooperative multilayer edge caching in an integrated satellite-terrestrial network was investigated, in which the base station cache, the satellite cache, and the gateway cache cooperatively provide content service for ground users. A three-layer cooperative caching model of the network was proposed, and two caching strategies were developed to reduce the communication delay. In [20], a cooperative distributed caching algorithm based on game theory in an IST network was proposed to reduce the content delivery delay while saving bandwidth resources.

Some of the aforementioned studies assumed that each deployed application instance can only serve a single application user; that is, the application instance is non-shareable. In studies that considered cached content to be shareable, most works did not account for the communication interference that may arise when a single instance on a satellite serves multiple users simultaneously. In practice, to reduce the resource deployment cost for satel-

lites, cost-effective content caching aims to lease minimal satellite storage resources while providing content services to the maximum number of application users within a given region. However, when edge servers need to serve an excessive number of users simultaneously, severe interference may occur, which can immediately degrade the achievable data transmission rate for application users [21]. In QoE-sensitive scenarios, insufficient data transmission rates make it difficult to ensure the perceived service quality for ground users. Perhaps the most related work is [22]. In [22], the multiple edge application deployment (MEAD) problem in the ground MEC environment is addressed, aiming to maximize application users' overall service quality at the minimum deployment cost while considering application shareability and communication interference. However, the MEC scenarios considered were limited to traditional terrestrial base stations and did not take into account the impact of the distance between the user and the edge server on the QoE. This factor becomes particularly critical in LEO satellite-enabled networks, where spatial dynamics can lead to significant variations in latency and link quality. Therefore, it is essential to design appropriate caching and service provisioning schemes that can effectively support heterogeneous content requests in such satellite-based edge computing environments.

Motivated by the above observations, this paper investigates LEO satellite-enabled MEC networks, where multiple LEO satellites equipped with caching capabilities act as edge servers to serve users with diverse content demands. The main contributions of this work are summarized as follows:

1. This paper characterizes the average QoE of users based on the effective transmission rate and adopts it as the performance metric. Note that the average QoE of users enables us to characterize the capability of our network in satisfying the users' heterogeneous content requests. Moreover, through analysis, we reveal that the average QoE of users is determined by the content cache decisions at the satellites, taking into account the interference at the user end.
2. To enhance the performance of LEO satellite-enabled MEC networks in satisfying users' content demands, we formulate an average user QoE maximization problem, subject to practical constraints on the satellites' caching and computation capabilities. To solve this non-convex problem, a two-stage content caching algorithm based on the divide-and-conquer and greedy strategies is designed. Via numerical results, we verify the effectiveness of our proposed algorithms through comparisons with various benchmarks, especially when the caching resources at the satellites are limited.

The remainder of this paper is organized as follows. Section 2 presents the system model and formulates the considered problem. Section 3 proposes the considered LEO satellite-enabled MEC network's content caching design. In Section 4, the numerical results are provided. Finally, Section 5 contains the conclusions.

2. System Model and Problem Formulation

2.1. Network Model

We consider an edge network composed of S LEO satellites, where all satellites act as base stations and are interconnected through a specific network topology. Each satellite is equipped with an edge server caching multiple content instances, thereby providing data services to M ground users, as illustrated in Figure 1.

In this paper, we set $\mathcal{S} = \{1, 2, \dots, S\}$ and $\mathcal{M} = \{1, 2, \dots, M\}$ to represent the nodes of LEO satellite edge servers (referred to as SESs) and the ground users, respectively.

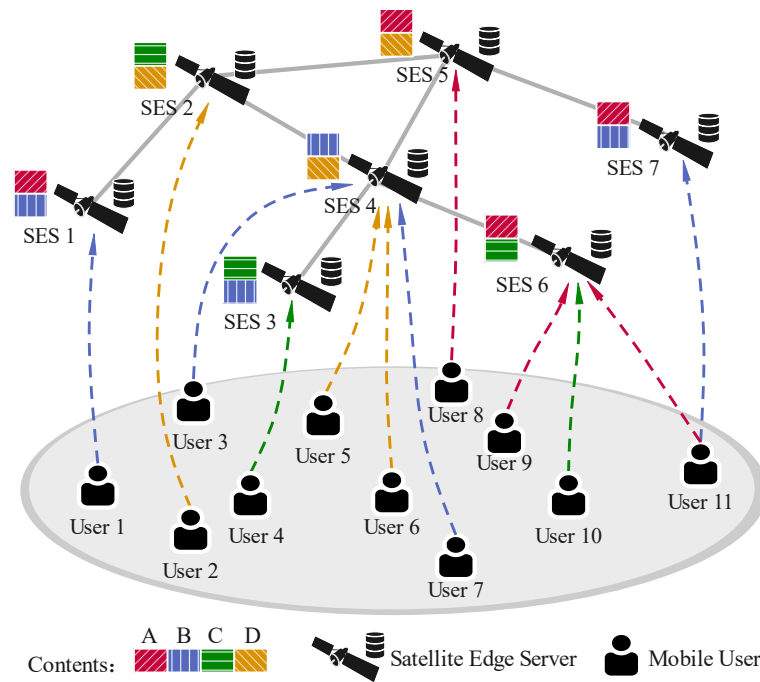


Figure 1. Illustration of our considered satellite-enabled edge networks, in which the users have heterogeneous content requests. Arrows of different colors represent different types of content that users need.

2.2. Cache Model

In this subsection, we detail the cache model of the SES nodes. The SES edge network considers a total of $|\mathcal{K}|$ types of content services corresponding to various mobile applications (hereafter referred to as “content”), denoted by the set $\mathcal{K} = \{1, 2, \dots, K\}$. Each type of content typically exists as a database or file instance and can be cached on multiple SES edge servers to provide services to ground users. Accordingly, each user m will request specific content data from the SES network. A binary variable $g_{m,k}$ is used to indicate whether user m requests content type k . It is worth noting that since a user’s requests for multiple application instances can equivalently be modeled as multiple virtual users each requesting a single content instance, for the sake of convenience, this paper assumes that each user requests only one specific content instance.

Each SES can choose to cache one or more content instances on its edge server. A binary variable $x_{k,s}$ is defined to indicate whether content instance k is cached on SES s , where $x_{k,s} = 1$ means SES s caches content instance k and $x_{k,s} = 0$ means SES s does not cache content instance k . The caching strategies of all SESs s regarding all K types of content instances are represented by a decision matrix $\mathbf{X}$. Each content instance requires a certain amount of storage resources when cached, denoted by c_k as the storage space required for content instance k . The storage capacity of the edge server equipped on SES s is denoted by St_s , which limits the maximum number of content instances that can be cached by SES s :

$$\sum_{k=1}^K x_{s,k} c_k \leq St_s, \forall s \in \mathcal{S} \quad (1)$$

Next, we present the process of handling the users’ requests.

2.3. User Access Model

In the network system considered in this paper, each SES has a certain coverage area. The matrix $\mathbf{A} = [\mu_{m,s}]_{M \times S}$ is used to represent the wireless coverage information of all SESs over the ground users, where $\mu_{m,s}$ is a binary variable. If $\mu_{m,s} = 1$, then this indicates

that user m is within the coverage area of SES s , and if $\mu_{m,s} = 0$, then this means that user m is outside the coverage area of SES s . Ground users can only access the relevant content data cached on the edge servers of the SES nodes that cover them. When a user is simultaneously covered by multiple SESs that cache the requested content, the user can choose to access the content data from any of these SESs.

It is worth emphasizing that data transmission between SESs introduces network latency, which is generally proportional to the number of communication hops involved. To reflect the latency sensitivity of ground users in the SES network, we assume in this work that users can only retrieve content either from the SES directly covering them or from one-hop neighboring SESs. While this constraint simplifies the system model, it is consistent with practical latency requirements. We note that with the adoption of more advanced data transmission technologies, such latency constraints could be further relaxed [23], although such considerations are beyond the scope of this paper.

Accordingly, a binary variable $b_{m,s}$ is defined to indicate whether user m can establish a wireless link with SES s and access cached data. The rule for determining the value of $b_{m,s}$ is defined as follows:

$$b_{m,s} = \begin{cases} 1, & \mu_{m,s} = 1 \text{ or } \mu_{m,s'} = 1, \forall s' \in \Phi(s) \\ 0, & \text{Otherwise} \end{cases} \quad (2)$$

where $\Phi(s)$ denotes the set of all SESs that are adjacent to SES s . When ground user m is covered by SES s (i.e., $\mu_{m,s} = 1$), the user can directly access the cached content data from SES s , meaning $b_{m,s} = 1$. Alternatively, if user m is covered by any SES $s' \in \Phi(s)$ (i.e., $\mu_{m,s'} = 1$), then user m can also indirectly access the content cached on SES s via network communication between SES s' and SES s .

2.4. Interference Model

Considering the sharing nature of content instances (i.e., multiple users may retrieve instance data from the same SES node), when a single SES s provides content services to too many ground users simultaneously, due to the typically limited spectral resources of LEO satellite constellations, significant co-channel interference may arise among ground users. This inter-user interference becomes more prominent as the number of ground user devices increases, reducing the actual data transmission rate between SES s and each user and further affecting the users' QoE for content services. Similar to [22,24], the following describes the communication interference model when multiple users share the same content instance. Let $\mathcal{M}_s$ denote the set of users that can access content data from SES s where their requested content is cached (i.e., $\mathcal{M}_s = \{m | b_{m,s} = 1, \forall m \in \mathcal{M}\}$). Then, for any user $m \in \mathcal{M}_s$, the effective transmission rate when receiving content data from SES s is expressed as follows:

$$r_{m,s} = W_s \log_2 \left(1 + \frac{P_m^{\text{tr}} h_{m,s}}{\sigma_0^2 + \sum_{m' \in \mathcal{M}_s \setminus \{m\}} P_{m'}^{\text{tr}} h_{m',s}} \right), \quad (3)$$

where W_u denotes the bandwidth allocated between SES s and user m , P_m^{tr} represents the wireless transmission power of ground user m , and σ_0^2 indicates the thermal noise power, while $h_{m,s}$ is the channel gain between user m and SES s , and its specific value varies depending on the particular SES and user. For the sake of concise discussion, this paper models $h_{m,s}$ based on the line-of-sight (LoS) channel, specifically

$$h_{m,s} = d_{m,s}^{-\eta}, \quad (4)$$

where $d_{m,s}$ and η represent the distance between user m and SES s and the path loss exponent of the wireless channel, respectively.

As previously described, due to the existence of communication links between satellites, users can access content data services from the one-hop neighboring SES nodes of the SES node that directly covers them. Suppose that user m is not within the coverage area of SES s' but is within the coverage of SES s . Then, the perceived transmission rate of user m when obtaining content data from SES s is determined by two factors: the data transmission rate between user m and SES s 's edge server and the data transmission rate between SES s' and SES s 's edge server. When SES s caches the content instance requested by user m and covers user m , the actual data reception rate $\gamma_{m,u}$ perceived by user m from SES s can be calculated as follows:

$$\gamma_{m,s} = \max\{\mu_{m,s}r_{m,s'}, \max_{s' \in \mathcal{M}_s \wedge \mu_{m,s'}} (\min\{r_{m,s'}, r_{s,s'}^{\text{SES}}\})\} \quad (5)$$

where $\gamma_{m,s}$ is the data transmission rate between SES s and user m , while $r_{s,s'}^{\text{SES}} = r_{s',s}^{\text{SES}}$ denotes the allocated bandwidth between two different SES nodes s and s' . It should be noted that in the SES network considered in this paper, the connection hop count between any two SES nodes is determined by the shortest connection path. Therefore, the actual bandwidth $r_{s,s'}^{\text{SES}}$ between SES s and s' is defined as the minimum value of the transmission bandwidths along the path.

2.5. Performance Metrics

The data rate corresponds to the quality of service level and may have varying impacts on the content service experience quality of application users. This paper assumes that the user's experience quality is strongly correlated with the actual data rate of the requested content instance. Specifically, as the actual data rate decreases, the user's experience quality degrades rapidly. In practical scenarios, since user m may be simultaneously covered by multiple neighboring SES nodes communicating with SES s , without loss of generality, this paper considers that user m will receive data service from the SES node capable of providing the highest data rate. Therefore, let Q_m denote the perceived QoE of user m regarding the content service in the considered SES edge network, which can be expressed as follows:

$$Q_m = \max_s \{1 - e^{-\alpha \gamma_{m,s}}\}, \forall s \in \mathcal{S} \quad (6)$$

where α denotes the sensitivity index of the user's experience quality to the data rate.

2.6. Problem Formulation

Given the previously described system, with a set of LEO satellite nodes equipped with edge servers $\mathcal{S}$, a set of ground users $\mathcal{M}$, and M corresponding content request demands, it is necessary to find a feasible SES content instance caching strategy $\mathbf{X}_{\mathcal{S} \times \mathcal{K}}$ that caches up to K types of content instances to provide data services to users. The objective of this work is to maximize the average QoE of all ground users $\bar{Q}$ while minimizing the SES network caching cost C . Our goal is mathematically formulated as follows:

$$\mathbf{P1:} \quad \max \bar{Q} = \frac{1}{M} \sum_{m=1}^M Q_m(\mathbf{X}) \quad (7)$$

$$\min C = \sum_{s=1}^S \sum_{k=1}^K x_{s,k} c_k$$

$$\text{s.t.} \quad g_{m,k} \in \{0, 1\}, \quad \forall m \in \mathcal{M}, \quad k \in \mathcal{K}, \quad (7a)$$

$$x_{s,k} \in \{0, 1\}, \quad \forall s \in \mathcal{S}, \quad k \in \mathcal{K}, \quad (7b)$$

$$\sum_{k=1}^K g_{m,k} \leq 1, \quad \forall m \in \mathcal{M}, \quad (7c)$$

$$\sum_{k=1}^K x_{s,k} c_k \leq St_s, \quad \forall s \in \mathcal{S}, \quad (7d)$$

$$\sum_{s=1}^S \mathcal{I}\{\gamma_{m,s} > 0\} = 1, \quad \forall m \in \mathcal{M}, \quad (7e)$$

$$\mathcal{I}\{Q_m > 0\} \leq \sum_{s=1}^S \sum_{k=1}^K g_{m,k} x_{s,k}, \quad \forall m \in \mathcal{M}. \quad (7f)$$

In Problem **P1**, the constraint in Equation (7a) specifies that each user has only two possible states regarding any content instance: either requesting it or not requesting it. The constraint in Equation (7b) describes that each SES can either choose to cache or not cache any specific content instance. The constraint in Equation (7c) indicates that each user can request at most one type of content. The constraint in Equation (7d) ensures that the total storage resources used by all cached content on each SES do not exceed its storage capacity limit. Finally, the constraints in Equation (7e,7f) state that each user's content request can be served by at most one SES, and all users' content requests must be fully satisfied.

We note that Problem **P1** in Equation (7) is mathematically intractable, due to the coupled different kinds of content cache variables in the objective function and non-convex constraints. Traditionally, this problem is solved using the exhaustive search method, which can be computation- and time-consuming, especially when the scale of the satellite network, the content number, and the number of users become relatively large. To resolve this issue, we propose a simple-yet-efficient design that can judiciously determine the content instance cache decisions at the satellites. The details of our proposed design will be presented in Section 3.

3. Proposed Satellite Edge Network Content Caching Design

In this section, we propose a novel design to address Problem **P1** in Equation (7). Specifically, we first reformulate Problem **P1** to simplify the problem. Then, we develop a two-stage optimization algorithm based on a greedy strategy, which iteratively solves the content caching problem for all SES nodes with low computational complexity.

3.1. A Two-Stage Content Caching Algorithm Based on Greedy Policies

To address Problem **P1**, it is necessary to tackle two main challenges: enhancing the average user-perceived QoE and reducing the storage space occupied by content instances cached on SES nodes. In practice, caching more types of content instances in the SES network is typically required to satisfy diverse user content requests, which conflicts with the goal of minimizing storage consumption. We note that Problem **P1** is a bi-objective optimization problem. Similar to the approach in [22], for simplicity, to express the joint optimization goal of minimizing the content caching cost and maximizing user experience quality in the simplest possible manner, we transform it into a single-objective optimization problem (Problem **P2**) by using the ratio between the total content caching consumption and the average user QoE:

$$\begin{aligned} \mathbf{P2}: \quad & \min \left[\sum_{m=1}^M Q_m(\mathbf{X}) \right]^{-1} \sum_{s=1}^S \sum_{k=1}^K x_{s,k} c_k \\ \text{s.t.} \quad & (7a) \sim (7f). \end{aligned} \quad (8)$$

This approach not only accounts for the contribution of both factors to the optimization objective but also makes the original problem more tractable. Nevertheless, we note that Problem **P2** in Equation (8) is still a mixed-integer linear programming problem which is NP-hard. Solving such a problem is computationally challenging if employing traditional centralized methods (e.g., generalized Benders decomposition [25] and Lagrangian decomposition [26]). To address this issue, we propose a low-complexity algorithm based on a divide-and-conquer strategy to iteratively address the content caching problem. Specifically, the algorithm first decomposes Problem **P2** into smaller-scale caching problems according to the types of content requests and then employs a conflict-merging strategy to optimize the caching decisions across the entire SES network.

Given any k th type of content instance, a feasible set of SESs capable of caching the current content instance can be obtained by searching, denoted as $\mathcal{S}_k = \{s | \forall s \in \mathcal{S} : St_s > c_k \wedge \exists m \in \mathcal{M}_s : g_{m,k} = 1\}$, along with a set of users requesting the k th type of content $\mathcal{M}_k = \{m | \forall m \in \mathcal{M} : g_{m,k} = 1\}$. For each SES with a sufficient storage capacity to cache the k th content instance, the corresponding set of users it can serve is denoted by $\mathcal{M}_k^s$, satisfying $\bigcup_{s=1}^S \mathcal{M}_k^s = \mathcal{M}_k$. When $\mathcal{M}_k^s \neq \emptyset$, this indicates that within the coverage area of SES s , there exist ground users requesting the k th content instance. In this case, SES s needs to cache one instance of the k th content. Therefore, a specific content caching scenario can be decomposed into multiple subproblems, where each subproblem focuses solely on the caching of a particular type of content instance on SES nodes.

As shown in Figure 2, the first subproblem only involves the caching issue of content A on SES 3, SES 4, and SES 5. Meanwhile, caching different types of content instances on the same SES may lead to conflicts. For example, when the total storage resource demand of the content A and content B instances exceeds the storage capacity of SES 4, both cannot be cached simultaneously on SES 4.

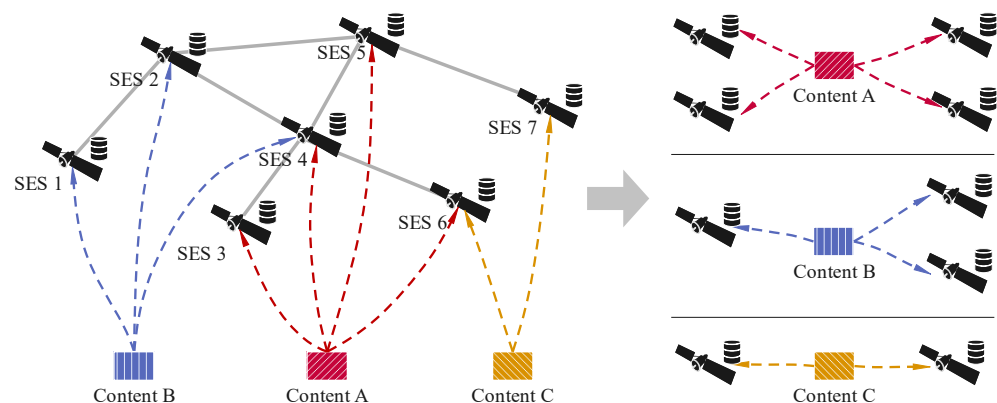


Figure 2. Illustration of the decomposition process for the SES content caching problem.

This paper proposes a scheme, namely the greedy partition-merge algorithm (GPMA), to iteratively address the caching problem. The GPMA consists of two stages: subproblem partitioning and conflict merging. Specifically, it first continuously searches for available SES nodes to pre-cache each content instance until all relevant user requests are fully satisfied. Then, by merging the caching schemes of all K types of content instances, a complete caching strategy is obtained to solve the content caching problem in the considered satellite network.

In the first stage of the GPMA, the content caching problem of the entire SES network is partitioned into smaller subproblems, aiming to iteratively pre-cache instances of each content on the SESs until the storage resource requirements of all content instances are satisfied within the SES network and ensuring that all user requests are served with the corresponding content. According to Equation (8), the caching decision problem for the k th content instance can be formulated as follows:

$$\min \left[\sum_{m=1}^M g_{m,k} Q_m(\mathbf{X}) \right]^{-1} \sum_{s=1}^S x_{s,k} c_k \quad (9)$$

The objective of Equation (9) is to find the caching strategy for the k th content instance such that the storage resources invested by the SES network are minimized while maximizing the perceived QoE for users requesting the k th content instance. Before introducing the caching strategy for the k th content instance, we first define $v_{s,k}$ as the effective unit value generated by caching the k th content instance on SES s :

$$v_{s,k} = \frac{1}{|\mathcal{M}_k^s|} \sum_{m \in \mathcal{M}_k^s} Q_m \quad (10)$$

To achieve the lowest possible content caching cost and the highest average user QoE, it is necessary to wisely select the target SES such that as many user requests for the k th content instance as possible are satisfied within the SES coverage area. Therefore, in each iteration of the algorithm, the key ideas for selecting the new target caching SES are as follows:

1. Give priority to select SESs that can generate a larger effective unit value for caching the content;
2. Give priority to select SESs that can satisfy more previously unmet user requests.

Given that our goal is to improve the average user's QoE while minimizing caching consumption, both of the prioritization principles above must be therefore jointly considered in the greedy selection process. We define the utility value [22,27] $R_{s,k}$ of the caching content k on SES s as the product of the effective unit value and the number of additional users that can be served by that SES s . According to Equation (10), the mathematical form of $R_{s,k}$ is as follows:

$$R_{s,k} = |\mathcal{M}_{s,k} - \mathcal{M}_k^*| \cdot v_{s,k}, \quad (11)$$

where $\mathcal{M}_k^* = \{m | m \in \mathcal{M}_k^s : x_{s,k} = 1, \forall s \in \mathcal{S}_k\}$ represents the set of users whose requests for content k have already been satisfied within the entire SES edge network, and it is initially an empty set. $|\mathcal{M}_{s,k} - \mathcal{M}_k^*|$ denotes the number of users belonging to set $\mathcal{M}_{s,k}$ but not belonging to set $\mathcal{M}_k^*$. It should be noted that during the iteration process of the algorithm, as more user requests for the k th content are satisfied (i.e., $|\mathcal{M}_k^*|$ increases), the utility value $R_{s,k}$ of caching the k th content on SES s will also dynamically change.

Next, we focus on introducing the second stage of the GPMA, namely merging the caching decisions of different content instances in the SES network, denoted by $X_1, X_2, \dots, X_K$, and resolving caching decision conflicts caused by SES storage capacity limitations. We use $\mathbf{X} = [\tilde{x}_{s,k}]_{S \times K}$ to represent the final merged SES network content caching decision, which is initially a zero matrix. At the same time, we denote G as the set of SESs whose caching conflicts have been resolved. At the beginning of the merging stage, the algorithm sorts all K types of content caching decisions $X_1, X_2, \dots, X_K$ in non-decreasing order based on the computed value of Equation (9) under specific strategies.

Meanwhile, for each caching decision X_k , $k \in \mathcal{K}$, the set of all SESs caching content k is obtained, denoted by G_k .

For any SES s , since the total storage resources required to cache multiple types of content may exceed the storage capacity of SES s , the caching decisions for different content instances on SES s may conflict. The concept of caching conflict is introduced as follows. Given a content instance $k \in \mathcal{K}$ and its corresponding SES caching strategy $X_k = [x_{s,k}]_{S \times K}$, let $V_s = \{k | k \in \mathcal{K} : \tilde{x}_{s,k} = 1\}$ represent the set of different content instances cached on SES s under the current merged caching strategy X . Then, if there exists SES $s \in G \cup G_k$ such that $\sum_{k' \in V_s \cup k} c_{k'} > St_s$, then a caching conflict is considered to exist on SES s ; otherwise, no caching conflict exists on SES s .

According to the above definition of caching conflict, we use $G'_k = \{s | s \in G \cap G_k : \sum_{k' \in V_s \cup k} c_{k'} > St_s\}$ to represent the set of SES nodes in which caching conflicts occur between G and G_k . Then, the algorithm greedily resolves the caching conflicts of content k on each SES $s \in G'_k$ in a non-increasing order based on the specific value of $v_{s,k}$. For each SES $s \in G'_k$, all conflicting caching strategies for content k are sorted in a non-decreasing order according to the value of $v_{s,k}/c_k$, meaning caching strategies with smaller $v_{s,k}/c_k$ values are given higher optimization priority. This prioritization ensures that when the storage occupancies c_k of two types of content are the same, the content with a larger unit value $v_{s,k}$ is prioritized for caching. Conversely, when the unit values $v_{s,k}$ are the same, the content with a smaller storage occupancy c_k is prioritized, thereby making full use of the SES's storage resources.

Next, to resolve the caching conflicts of the content k on each SES $s \in G'_k$, the GPMA selects a content $k' \in V_s$ with the smallest $v_{s,k'}/c_{k'}$ ratio from V_s . Then, it compares the current content k to be cached and the already cached content k' and attempts to replace SES s with another SES s' (or SES s'') from its neighboring SES set $\Phi(s)$ based on the maximum value among $v_{s,k'}$, $v_{s',k}$ (or $v_{s'',k}$). If $v_{s,k} + v_{s'',k'} \geq v_{s',k} + v_{s,k'}$, then the content k' originally cached on SES s will be migrated to SES s'' , and the caching strategy matrix $X_{k'}$ will be updated; otherwise, content k will be cached on SES s' , and the caching strategy matrix X_k will be updated, thereby resolving the caching conflict of SES s for content k . The algorithm repeats the above process until all caching conflicts of content k on SESs $s \in G'_k$ are resolved. Once the caching strategies of all K types of content are merged, the entire optimization process terminates.

The overall process of the GPMA is shown in Algorithm 1, where steps 1–14 are the partition phase and steps 15–25 are the merge phase.

3.2. Algorithm Convergence and Complexity Analysis

The computational complexity of Algorithm 1 arises from two sub-processes: the caching partition phase and the caching conflict merging phase. In the first phase (steps 1–14), for any content $k \in \mathcal{K}$, traversing all SESs $s \in S_k$ has a time complexity of $\mathcal{O}(S)$, and finding the SES with the minimum utility value in the SES set S_k also has a time complexity of $\mathcal{O}(S)$. Therefore, the overall time complexity of obtaining the caching partition strategies X_k , $\forall k \in \mathcal{K}$ for all K types of content is $\mathcal{O}(KS)$. In the second phase (steps 15–25), sorting all K types of content has a time complexity of $\mathcal{O}(K \log K)$. For each content $k \in \mathcal{K}$, in the worst case, caching conflicts occur on all SESs, and the most complex network scenario is when each SES s has $S - 1$ neighboring SESs. Therefore, for all K types of content, merging their caching conflict strategies has a time complexity of $\mathcal{O}(KS)$. In summary, the overall computational complexity of Algorithm 1 is $\max\{\mathcal{O}(K \log K), \mathcal{O}(KS)\}$.

Regarding convergence, based on the analysis above, since the highest computational complexity of Algorithm 1 is polynomial, both phases of the algorithm can be completed in a finite number of steps, thus guaranteeing the convergence of Algorithm 1.

Algorithm 1: Two-stage content caching algorithm based on greedy policy.

Input: Users' requests, the states of the SES network.

```

1 Initialize  $X, G, X_k, G_k$  and  $\mathcal{M}_k^*$  ( $\forall k \in \mathcal{K}$ ) as zero matrices.
2 for each content type  $k \in \mathcal{K}$  do
3   repeat
4     for each SES  $s \in \mathcal{S}_k$  do
5       if  $\mathcal{M}_{s,k} \subseteq \mathcal{M}_k^*$  then
6         Remove  $s$  in the SES set  $\mathcal{S}_k$ .
7       else
8         Calculate the unit value  $v_{s,k}$  and utility value  $R_{s,k}$  resulting from the
          caching content  $k$  on  $s$ ;
9       end
10    end
11    Obtain the SES  $s$  that maximizes  $R_{s,k}$  ( $s \in \mathcal{S}_k$ ), and update  $X_k$ :  $x_{s,k} \leftarrow 1$ ;
12    Add SES  $s$  to set  $\mathcal{M}_k^*$  and remove it from  $\mathcal{S}_k$ ;
13  until  $\mathcal{M}_k^* = \mathcal{M}_k$ ;
14 end
15 for each  $k \in \mathcal{K}$ , in non-increasing order of Equation (9) do
16   for each SES  $s \in G'_k$  in non-decreasing order of  $v_{s,k}$  do
17     Obtain the content  $k'$  that maximizes  $v_{s,k}/c_k$  in the set  $V_s$  of the already
      cached contents of  $s$ ; if  $v_{s,k} + v_{s'',k'} \geq v_{s',k} + v_{s,k'}$  then
18       Update the decision matrix  $X_{k'}$ :  $x_{s,k'} \leftarrow 0, x_{s'',k'} \leftarrow 1$ ;
19     else
20       Update the decision matrix  $X_k$ :  $x_{s,k} \leftarrow 0, x_{s',k} \leftarrow 1$ ;
21     end
22   end
23   Merge the updated decision matrix  $X_k$  into  $X$ ;
24   Merge the set of SESs that all satisfy  $x_{s,k} = 1$  into  $G$ ;
25 end
26 return Caching strategy  $X$  for merged content of all  $K$  types.

```

4. Numerical Results

In this section, we provide numerical results to validate the effectiveness of our proposed algorithms. Without loss of generality, we considered a simulating area of $2000 \text{ km} \times 2000 \text{ km}$, in which S SESs and M users were deployed. The locations of the users were randomly generated according to the Poisson point process (PPP). The altitude and coverage radius of the SESs were set to 400 km. Without loss of generality, we uniformly divided the fixed simulation area into a 4×4 grid and deployed a total of 16 satellites, each located at the center of a grid cell [13]. This set-up enabled relatively uniform coverage of the ground area by the satellite network. In addition, we considered K different content types, and each user randomly uniformly requested these data of contents with a probability $1/K$. To emulate the QoE model, we referenced the curve parameters in [28], and the value of α was set to 0.5. The simulation performances were obtained through averaging over 1000 simulations. (Although not shown, we note that the average performances of our proposed design and other benchmark schemes became stabilized when the number of simulations was larger than 500. As such, setting the number of simulations to 1000 was sufficient.) Unless otherwise specified, the simulation parameters used throughout the simulations are listed in Table 1.

That aside, several baseline schemes were set up during the simulation process to better verify the effectiveness of the proposed GPMA algorithm, specifically the following:

1. Non-cooperation caching [13] (NCC): All SESs ignore potential communication with other SESs and focus solely on optimizing the QoE for ground users within their own wireless coverage areas. Each SES independently determines its content caching strategy.
2. Coverage prioritization caching [29] (CPC): For each content instance, the algorithm greedily selects the SES that can provide the content to the largest number of unsatisfied requesting users, caching the content instance there until all user requests are fulfilled.
3. Random caching [22] (RC): SESs randomly cache various content instances until the content requests of all users are covered.

Table 1. List of simulation parameters.

Parameters	Values
Number of SESs, S	16
Number of ground users, M	[50, 100]
Number of content type varieties, K	[2, 12]
SES storage space, St_s	2 GB
Size of instance of content k , c_k	[200, 800] MB
Wireless link rate between SESs, r^{SES}	40 MB/s
Wireless bandwidth of subchannel, W	1 MHz
Signal transmission power of user, P_m^{tr}	1 W
Noise power, σ_0^2	−110 dBm

In Figure 3, we can observe that as the number of content types K increased, the average user QoE $\bar{Q}$ of all four schemes showed a decreasing trend. This indicates that for a fixed number of SESs and ground users, the increase in content service types forced the SES network to cache a greater variety of content instances to meet user demands. This significantly raised the probability of multiple content instances competing for storage resources on the same SES, making it more challenging to cache the requested content for users. We can see that the GPMA scheme consistently achieved a better average user QoE $\bar{Q}$ compared with the other schemes in most cases, with the performance gap becoming more pronounced as the number of content types increased. This demonstrates the GPMA scheme's superior ability to efficiently utilize SES storage resources for high-quality content data delivery.

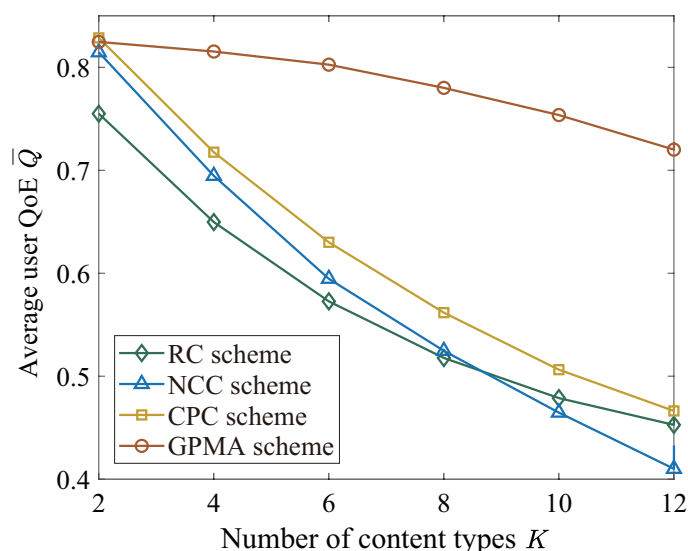


Figure 3. Average user QoE versus the number of content types with $\alpha = 0.5$, $S = 16$, and $M = 75$.

Additionally, it is worth noting that as the number of content types K grew, the NCC scheme experienced the most noticeable drop in QoE performance due to its disregard for potential cooperation between neighboring SESs, leading to lower utilization of storage space when dealing with a large variety of content instances.

Figure 4 shows the storage resource requirements under the four caching schemes. As the number of content types K increased, more content instances needed to be cached in the SES network, and the overall caching cost C of all schemes rose accordingly. It can be seen that the GPMA scheme's caching cost was significantly lower than those of the other three schemes. Meanwhile, RC, CPC, and NCC maintained consistently higher caching costs regardless of content variety. This is because the GPMA is designed to satisfy multiple users' requests with the minimal possible storage cost and seeks the most efficient caching strategy by leveraging cooperation among multiple neighboring SESs.

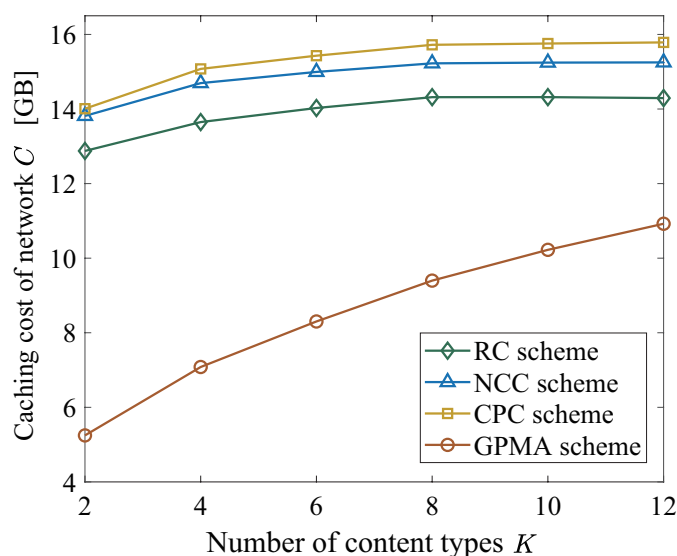


Figure 4. Caching cost of a network versus the number of content types with $\alpha = 0.5$, $S = 16$, and $M = 75$.

Figure 5 illustrates the average user QoE $\bar{Q}$ achieved by different caching strategies under varying numbers of ground users M . The figure also shows how the overall caching cost changed as the number of users increased. As M increased, the average user QoE $\bar{Q}$ of the three baseline schemes gradually decreased, mainly due to the increased user interference caused by SESs serving more users simultaneously. In contrast, the GPMA scheme maintained a relatively high $\bar{Q}$ by fully accounting for communication interference when optimizing content caching strategies. It is worth noting that as M increased, the average user QoE $\bar{Q}$ under the GPMA initially showed an upward trend. This is because the GPMA leverages the redundant storage resources of multiple SESs to alleviate interference among multiple users.

Aside from that, Figure 6 examines the caching cost C incurred by each strategy under different numbers of ground users M . It can be observed that as M increased, the overall caching cost C of all four schemes rose accordingly. This is due to the SES network needing to cache a wider variety of content instances locally to meet the increased demand. Notably, the GPMA achieved the lowest SES storage resource usage, highlighting the practical significance of the proposed algorithm in reducing caching costs and improving SES storage resource utilization efficiency.

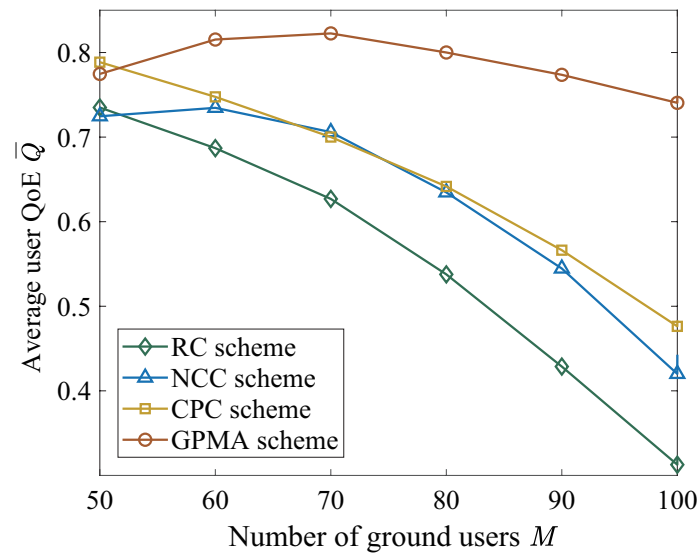


Figure 5. Average user QoE versus the number of ground users with $\alpha = 0.5$, $S = 16$, and $K = 5$.

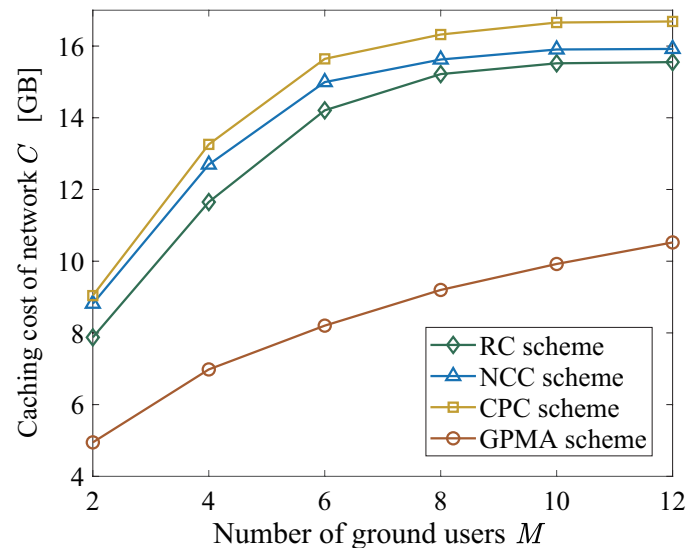


Figure 6. Caching cost of a network versus the number of ground users with $\alpha = 0.5$, $S = 16$, and $K = 5$.

5. Conclusions

In this paper, we considered a satellite-assisted edge network where multiple cache-enabled LEO satellites were deployed to support high-bandwidth data services and satisfy content requests generated by the ground users. We assumed that the effective transmission rate significantly impacted users' content experience quality. Accordingly, we adopted the average user QoE as the performance metric to evaluate the capability of our considered network in satisfying the users' differentiated content requests, while considering communication interference among multiple users. We proposed a simple-yet-efficient content caching algorithm based on a greedy strategy, which aims to optimize the caching decisions for multiple content instances in the satellite edge network to maximize the average user experience quality. The numerical results demonstrated the superior performance of the proposed scheme over various benchmark schemes, particularly under limited satellite caching and computation resources. In addition to significantly improving the average user

QoE, our approach effectively reduced the caching costs, highlighting its practical value in resource-constrained satellite edge environments.

Author Contributions: Conceptualization, J.H. and Y.Z.; methodology, J.H.; validation, Y.M. and Q.W.; writing—original draft preparation, Y.Z.; writing—review and editing, J.H.; visualization, Y.M.; supervision, Q.W.; project administration, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

$\mathcal{S}$	Set of SESs
$\mathcal{M}$	Set of ground users
$\mathcal{K}$	Set of content types
$g_{m,k}$	Indicator of whether content k is requested by user m
$x_{s,k}$	Indicator of whether SES s has content k in its cache
c_k	Size of each content instance
St_u	Storage capacity of SES
$\mu_{m,s}$	Indicator of the wireless coverage of a user-SES pair
$b_{m,s}$	Indicator of whether user m can communicate with s
$\Phi(s)$	Set of SESs that are adjacent to SES s
$\mathcal{M}_s$	Set of users that can access the content from s
$r_{m,s}$	Effective transmission rate when receiving content from s
$h_{m,s}$	Channel gain between user m and SES s
$\gamma_{m,s}$	Data transmission rate between SES s and user m
Q_m	Perceived QoE of user m
$\bar{Q}$	Average QoE of all ground users
C	Caching cost of the SES network
$\mathcal{S}_k$	Set of SESs capable of caching content k
$\mathcal{M}_k$	Set of users requesting content k
$\mathcal{M}_k^s$	Set of users requesting content k from SES s
$v_{s,k}$	Unit value of caching content k on SES s
$R_{s,k}$	Utility value of caching content k on SES s
$\mathcal{M}_k^*$	Set of users whose requests have been satisfied
$\mathbf{X}_k$	Decisions of caching content k
$\mathbf{X}$	Decisions of caching all K types of content
G	Set of SESs that no longer have caching conflicts
G_k	Set of all SESs where caching content k is obtained
G'_k	Set of SESs that have conflicts between G and G_k

References

1. Daurembekova, A.; Schotten, H.D. Opportunities and Limitations of Space-Air-Ground Integrated Network in 6G Systems. In Proceedings of the 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Toronto, ON, Canada, 5–8 September 2023. [\[CrossRef\]](#)
2. Azari, M.M.; Solanki, S.; Chatzinotas, S.; Kotheli, O.; Sallouha, H.; Colpaert, A. Evolution of Non-Terrestrial Networks From 5G to 6G: A Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2633–2672. [\[CrossRef\]](#)
3. Su, Y.; Liu, Y.; Zhou, Y.; Yuan, J.; Cao, H.; Shi, J. Broadband LEO Satellite Communications: Architectures and Key Technologies. *IEEE Wireless Commun.* **2019**, *26*, 55–61. [\[CrossRef\]](#)

4. Massoglia, P.; Pozesky, M.; Germana, G. The use of satellite technology for oceanic air traffic control. *Proc. IEEE* **1989**, *77*, 1695–1708. [\[CrossRef\]](#)
5. Zhang, B.; Wu, Y.; Zhao, B.; Chanussot, J.; Hong, D.; Yao, J.; Gao, L. Progress and Challenges in Intelligent Remote Sensing Satellite Systems. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1814–1822. [\[CrossRef\]](#)
6. Casoni, M.; Grazia, C.A.; Klapez, M.; Patriciello, N.; Amditis, A.; Sdongos, E. Integration of satellite and LTE for disaster recovery. *IEEE Commun. Mag.* **2015**, *53*, 47–53. [\[CrossRef\]](#)
7. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [\[CrossRef\]](#)
8. Yang, L.; Cao, J.; Liang, G.; Han, X. Cost Aware Service Placement and Load Dispatching in Mobile Cloud Systems. *IEEE Trans. Comput.* **2016**, *65*, 1440–1452. [\[CrossRef\]](#)
9. He, X.; Wang, S.; Wang, X.; Xu, S.; Ren, J. Age-Based Scheduling for Monitoring and Control Applications in Mobile Edge Computing Systems. In Proceedings of the IEEE INFOCOM 2022–IEEE Conference on Computer Communications, London, UK, 2–5 May 2022. [\[CrossRef\]](#)
10. Kim, T.; Sathyanarayana, S.D.; Chen, S.; Im, Y.; Zhang, X.; Ha, S. MoDEMS: Optimizing Edge Computing Migrations for User Mobility. In Proceedings of the IEEE INFOCOM 2022–IEEE Conference on Computer Communications, London, UK, 2–5 May 2022. [\[CrossRef\]](#)
11. Farhadi, V.; Mehmeti, F.; He, T.; La Porta, T.F.; Khamfroush, H.; Wang, S. Service Placement and Request Scheduling for Data-Intensive Applications in Edge Clouds. *IEEE/ACM Trans. Netw.* **2021**, *29*, 779–792. [\[CrossRef\]](#)
12. Xu, J.; Chen, L.; Zhou, P. Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks. In Proceedings of the IEEE INFOCOM 2018–IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018. [\[CrossRef\]](#)
13. Zhao, Y.; Liu, C.; Hu, X.; He, J.; Peng, M.; Ng, D.W.K. Joint Content Caching, Service Placement, and Task Offloading in UAV-Enabled Mobile Edge Computing Networks. *IEEE J. Sel. Areas Commun.* **2025**, *43*, 51–63. [\[CrossRef\]](#)
14. Gao, Y.; Guan, H.; Qi, Z.; Hou, Y.; Liu, L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *J. Comput. Syst. Sci.* **2013**, *79*, 1230–1242. [\[CrossRef\]](#)
15. Liu, Y.; Mao, Y.; Shang, X.; Liu, Z.; Yang, Y. Distributed Cooperative Caching in Unreliable Edge Environments. In Proceedings of the IEEE INFOCOM 2022–IEEE Conference on Computer Communications, London, UK, 2–5 May 2022. [\[CrossRef\]](#)
16. Xie, R.; Tang, Q.; Wang, Q.; Liu, X.; Yu, F.R.; Huang, T. Satellite-Terrestrial Integrated Edge Computing Networks: Architecture, Challenges, and Open Issues. *IEEE Netw.* **2020**, *34*, 224–231. [\[CrossRef\]](#)
17. Li, Q.; Wang, S.; Ma, X.; Sun, Q.; Wang, H.; Cao, S. Service Coverage for Satellite Edge Computing. *IEEE Internet Things J.* **2022**, *9*, 695–705. [\[CrossRef\]](#)
18. Pasteris, S.; Wang, S.; Herbster, M.; He, T. Service Placement with Provable Guarantees in Heterogeneous Edge Computing Systems. In Proceedings of the IEEE INFOCOM 2019–IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 514–522. [\[CrossRef\]](#)
19. Zhu, X.; Jiang, C.; Kuang, L.; Zhao, Z. Cooperative Multilayer Edge Caching in Integrated Satellite-Terrestrial Networks. *IEEE Trans. Wireless Commun.* **2022**, *21*, 2924–2937. [\[CrossRef\]](#)
20. Tang, J.; Li, J.; Chen, X.; Xue, K.; Zhang, L.; Sun, Q. Cooperative Caching in Satellite-Terrestrial Integrated Networks: A Region Features Aware Approach. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10602–10616. [\[CrossRef\]](#)
21. Han, P.; Liu, Y.; Guo, L. Interference-Aware Online Multicomponent Service Placement in Edge Cloud Networks and its AI Application. *IEEE Internet Things J.* **2021**, *8*, 10557–10572. [\[CrossRef\]](#)
22. Zhao, L.; Tan, W.; Li, B.; He, Q.; Huang, L.; Sun, Y. Joint Shareability and Interference for Multiple Edge Application Deployment in Mobile-Edge Computing Environment. *IEEE Internet Things J.* **2022**, *9*, 1762–1774. [\[CrossRef\]](#)
23. Li, B.; He, Q.; Cui, G.; Xia, X.; Chen, F.; Jin, H. READ: Robustness-Oriented Edge Application Deployment in Edge Computing Environment. *IEEE Trans. Serv. Comput.* **2022**, *15*, 1746–1759. [\[CrossRef\]](#)
24. Cui, G.; He, Q.; Chen, F.; Jin, H.; Yang, Y. Trading off Between Multi-Tenancy and Interference: A Service User Allocation Game. *IEEE Trans. Services Comput.* **2022**, *15*, 1980–1992. [\[CrossRef\]](#)
25. Geoffrion, A.M. Generalized benders decomposition. *J. Optim. Theory Appl.* **1972**, *10*, 237–260. [\[CrossRef\]](#)
26. Billionnet, A.; Soutif, É. An exact method based on Lagrangian decomposition for the 0–1 quadratic knapsack problem. *Eur. J. Oper. Res.* **2004**, *157*, 565–575. [\[CrossRef\]](#)
27. Tan, W.; Zhao, L.; Li, B.; Xu, L.; Yang, Y. Multiple Cooperative Task Allocation in Group-Oriented Social Mobile Crowdsensing. *IEEE Trans. Services Comput.* **2022**, *15*, 3387–3401. [\[CrossRef\]](#)

28. Fiedler, M.; Hossfeld, T.; Tran-Gia, P. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network* **2010**, *24*, 36–41. [[CrossRef](#)]
29. He, T.; Khamfroush, H.; Wang, S.; La Porta, T.; Stein, S. It's Hard to Share: Joint Service Placement and Request Scheduling in Edge Clouds with Sharable and Non-Sharable Resources. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 365–375. [[CrossRef](#)]

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.