Service Coverage for Satellite Edge Computing

Qing Li, Shangguang Wang, *Senior Member, IEEE*, Xiao Ma, Qibo Sun, Houpeng Wang,
Suzhi Cao, and Fangchun Yang, *Senior Member, IEEE*

***Abstract*—Recently, increasing investments in satellite-related technologies make the low earth orbit (LEO) satellite constellation a strong complement to terrestrial networks. To mitigate the limitations of the traditional satellite constellation "bent-pipe" architecture, satellite edge computing (SEC) has been proposed by placing computing resources at the LEO satellite constellation. Most existing works focus on space-air-ground integrated network architecture and SEC computing framework. Beyond these works, we are the first to investigate how to efficiently deploy services on the SEC nodes to realize robustness aware service coverage with constrained resources. Facing the challenges of spatial-temporal system dynamics and service coverage-robustness conflict, we propose a novel online service placement algorithm with a theoretical performance guarantee by leveraging Lyapunov optimization and Gibbs sampling. Extensive simulation results show that our algorithm can improve the service coverage by 4.3× compared with the baseline.**

***Index Terms*—Satellite edge computing (SEC), service coverage, space-air-ground integrated networks.**

I. Introduction

With the coming of the Internet-of-Things (IoT) era, massive devices all over the world need to be connected in the future. It is estimated that there will be more than 30 billion IoT devices deployed worldwide by the end of 2025 [1]. However, the terrestrial network covers only about 20% of the total land area [2]. Moreover, the terrestrial network is vulnerable to natural disasters, such as floods, earthquakes, tsunamis. An emerging technology, the low earth orbit (LEO) satellite constellation can provide ubiquitous access. Commercial efforts such as OneWeb, SpaceX Starlink, and Space Norway deploy large satellite constellations to LEO with a low per-device cost. They have secured the RF spectrum from the federal communications commission for their constellations [3]–[6]. Such systems have advantages in applications such as precision agriculture, environmental monitoring, disaster relief, and humanitarian assistance. Increasing investments in satellite-related technologies make the LEO satellite constellation a strong complement to terrestrial networks and future 5G/6G communications [7], [8]. The traditional LEO satellite constellations cannot provide computing service directly because the computing power is weak and there is no unified service framework and service interface. Most existing satellite systems adopt a "bent-pipe" architecture, where ground stations send commands to orbit and satellites reply with raw data. As the constellation population increases, the bent-pipe architecture breaks down because of limited link availability and bitrate bottlenecks. These limitations of bent-pipe architecture motivate the satellite edge computing (SEC) technique by placing computing resources at the LEO satellite constellation [9].

Most existing works related to SEC focus on space-air-ground integrated network architecture [7], [8], [10]–[17] and SEC framework [9], [18]–[21]. These works are the research foundation for service placement and computation offloading [22]. A few works focus on computation offloading to LEO satellites [2], [23]–[25]. They assume that services are already placed merely concentrate on the offloading strategies, while the offloading performance cannot be guaranteed if services are not available or service robustness is poor. While service placement has been investigated extensively in mobile edge computing [26]–[34], these works cannot be applied to SEC scenarios directly because of the unique characteristics of SEC systems, such as highly dynamic network topology.

To guarantee robustness-aware *service coverage* with constrained resources, this article studies how to efficiently deploy services on SEC nodes. For a single service, service coverage is defined as the user request number that can access the service, and service robustness is defined as the user request number that can access more than one service copies deployed on different SEC nodes. Services deployed in the volatile SEC environment are is vulnerable to events such as bandwidth fluctuations, insufficient computing resources, software exceptions/hardware faults, and physical satellite damage. Service failures may decline user quality of experience if they are not covered by any other service copies, especially for latency-sensitive services. How to achieve robustness-aware service coverage in the volatile SEC environment is challenging.

The first challenge is how to adapt to spatial-temporal system dynamics. On one hand, the core topology of SEC systems is inherently dynamic. The high-speed periodic motion of the satellite network causes dynamic channel

Manuscript received February 1, 2021; revised April 21, 2021; accepted May 26, 2021. Date of publication May 31, 2021; date of current version December 23, 2021. This work was supported in part by Key-Area Research and Development Program of Guangdong Province under Grant 2020B010164002; in part by NSFC under Grant 62032003 and Grant 61922017; and in part by the Open Research Fund of Key Laboratory of Space Utilization, Chinese Academy of Sciences under Grant LSUKFJJ-2019-03. *(Corresponding author: Suzhi Cao.)*

Qing Li, Xiao Ma, Qibo Sun, and Fangchun Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: q_li@bupt.edu.cn; maxiao18@bupt.edu.cn; qbsun@bupt.edu.cn; fcyang@bupt.edu.cn).

Shangguang Wang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: sgwang@bupt.edu.cn).

Houpeng Wang and Suzhi Cao are with the Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China (e-mail: wanghoupeng19@mails.ucas.ac.cn; caosuzhi@csu.ac.cn).

Digital Object Identifier 10.1109/JIOT.2021.3085129

conditions and changing connections with ground users and other satellites, leading to varying service coverage and robustness. On the other hand, the distribution of user requests varies temporally and spatially varied, e.g., the higher densities at cities in daytimes. The second challenge is how to balance service coverage and service robustness, which are conflict (see Section III-C). To improve service coverage, we need to deploy service copies (of a single service) on as many SEC nodes to cover as many user requests. To improve service robustness, we need to deploy service copies on SEC nodes overlapping with each other. How to tradeoff service coverage and robustness under a limited placement budget is challenging.

In this article, we propose a novel online service placement algorithm for SEC to maximize robustness aware service coverage with limited placement and operational budget. We first model service coverage and service robustness and then formulate the problem as an integer programming problem, which is proven to be an NP-hard problem. In particular, we seek to maximize the long-term average service coverage under the constraints of service placement and operational budgets which is predefined by service providers. By applying Lyapunov optimization technique, our algorithm transforms the long-term optimization problem into many real-time ones and then makes service placement decisions without any future information (e.g., service demand dynamics). In each real-time optimization problem, we use the Gibbs sampling method to achieve a near-optimal service placement decision. We prove that the performance loss of our algorithm is theoretically bounded.

The contributions are summarized as follows.

1) To the best of our knowledge, we are the first to study dynamic service coverage in SEC, considering SEC characteristics, user demand, service placement and operational costs.
2) We propose a novel online service placement algorithm with theoretical performance guarantees leveraging Lyapunov optimization and Gibbs sampling.
3) We conduct extensive simulations to evaluate the performance of our algorithm. The simulation results show that our algorithm can improve the service coverage by 4.3× compared with the baseline.

The remainder of this article is organized as follows. Section II reviews the related work. In Section III we analyze the system model and formulate the problem. In Sections IV and V, we develop the algorithm and illustrate simulation results. Finally, we conclude our work in Section VI.

II. Related Work

Network Architecture: Existing works mainly focus on the architecture of the space-air-ground network, which interconnects satellites, aerial platforms, as well as terrestrial communication systems. Boero *et al.* [7], Giambene *et al.* [11], Zhang *et al.* [12], and Shi *et al.* [13] investigated the integration of satellite networks and 5G networks leveraged by the software-defined network and network function virtualization technologies. Tang [15] proposed a software-defined network-based network architecture and manage resources in satellite-terrestrial networks in the layered and on-demand way. Varasteh *et al.* [16] proposed a reconfigurable service provisioning framework based on the service function chain for space-air-ground integrated networks. Some works [35]–[37] highlight the networking challenges in LEO networks. Besides, several works focus on SEC framework [18]–[21]. Bhattacherjee *et al.* [18] presented the opportunities and challenges of in-orbit computing. To address the limitations of the bent-pipe architecture, Denby and Lucia [19] proposed an orbital edge computing architecture to support edge computing at nanosatellite. Yan *et al.* [20] presented the system framework of SEC and its resource platform, where the hardware is based on an embedded platform and the software is decomposed into micro-services. Xie *et al.* [21] studied the satellite-terrestrial integrated edge computing networks, utilizing edge computing to improve the resource utilization of the satellite-terrestrial network. These network architecture works are the research foundation for the service placement and computation offloading works.

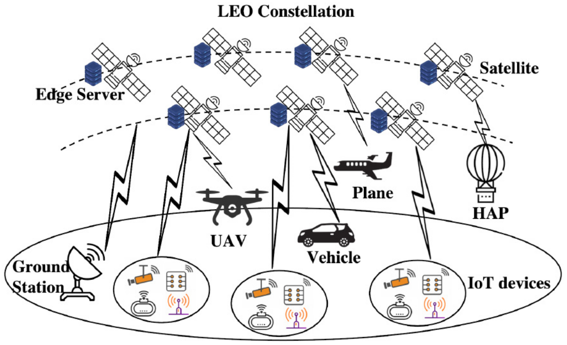


Fig. 1. SEC system model.

Computation Offloading in SEC: Wang *et al.* [23] proposed an offloading algorithm to optimize both energy consumption and latency for satellite-terrestrial networks with double edge computing. Du *et al.* [24] proposed a software-defined network-based architecture to support spectrum management and traffic offloading in satellite-terrestrial networks. Cheng *et al.* [25] presented a flexible joint communication and computation satellite-terrestrial integrated network framework to provide powerful computing services to IoT users. All these work study static service placement or computation offloading for a specific ground area while we study global on-demand service coverage considering network dynamics and service robustness.

III. System Model and Problem Formulation

A. SEC Network Overview

We consider an LEO constellation with a total number of N satellites denoted by $\mathcal{S} = \{1, \ldots, i, \ldots, N\}$. The satellite constellation orbits can be divided into polar orbits and inclined orbits. Polar orbits provide world coverage while inclined orbits provide better revisit of satellites over populated areas near the equator. Our model fits both kinds of constellations. In constellations, intersatellite links (ISLs) are

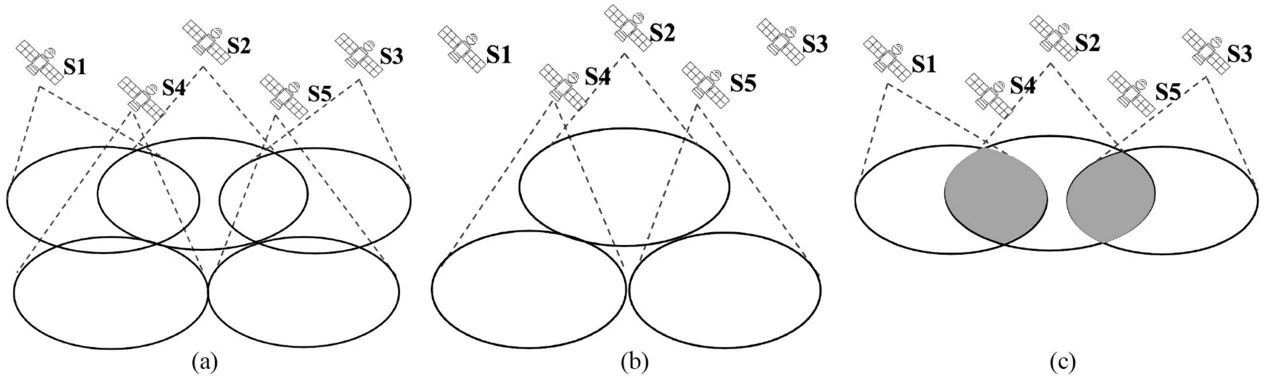


Fig. 2. Motivating example to illustrate the conflict between service coverage and robustness. (a) Scenario. (b) Maximizing service coverage. (c) Maximizing service robustness.

point-to-point links between satellites. A well-known topology forms a +Grid connectivity pattern where each satellite connects to two adjacent satellites in the same orbit, and two satellites in adjacent orbits [38]. Despite there is substantial uncertainty about whether or not ISLs will be deployed, the use of laser ISLs can achieve high network capacities (more than 100 Gb/s), which is appealing [36], [39]. In this article, we adopt +Grid ISL topology in the constellation. Note that we assume that user devices can communicate directly with satellites in their view without any gateway intervention [40]. And careful frequency management alleviates interuser interference so that each satellite can connect to multiple devices simultaneously [41], [42]. Each SEC node is deployed with edge computational and storage capability to provide computing services to end-users [19]. We name each satellite with edge servers as SEC node. As shown in Fig. 1, ground or aerial users such as base stations, IoT devices, vehicles, and planes, can directly request services deployed in SEC nodes [43]. In this article, we investigate dynamic service placement to achieve robustness aware service coverage in SEC. We first give an intuitive example to illustrate the conflict between service coverage and service robustness in Fig. 2. It can be seen that service coverage and service robustness are two conflict metrics. In the following, we model service placement decisions, service coverage, and service robustness formally.

B. Service Placement Decisions

We assume the time is discretized into time slots $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$. Each time slot is a service placement decision round. SEC nodes can create execution environments, e.g., VM or Container to place services that could benefit from SEC. These potential services include content distribution and multiuser interaction services such as multiuser gaming, co-immersion, and collaborative music [18]. Different services are heterogeneous in terms of computing resource requirements. There are K types of services indexed by $\mathcal{F} = \{1, \ldots, k, \ldots, K\}$. We denote the dynamic service placement decision with a binary variable $x_{ik}(t) \in \{0, 1\}$. If the service k is placed on SEC node i at time slot t then $x_{ik}(t) = 1$, otherwise, $x_{ik}(t) = 0$. The placement decision of all SEC nodes can be denoted by $\mathbf{x}(t) = (\mathbf{x}_1(t), \ldots, \mathbf{x}_i(t), \ldots, \mathbf{x}_N(t))$, $\mathbf{x}_i(t) = (x_{i1}, \ldots, x_{ik}, \ldots, x_{iK})$. Considering placement cost, each service k can be copied on at most B_k SEC nodes in the long term. In addition, not all services can be placed at an SEC node simultaneously because of the limited computing resources at the SEC node. We mainly consider that the processor capacity constraint at SEC nodes. The computation capacity requirement of service k is denoted by p_k and the computation capacity of SEC node i is denoted by P_i. To satisfy the computation capacity constraint, $\sum_{k \in \mathcal{F}} p_k x_{ik}(t) \leq P_i$ holds true. Notice that service reconfiguration (deploying and canceling) incurs an operational cost in the dynamic service placement. To model the operational cost, we use $x_{ik}(t-1) \oplus x_{ik}(t)$ to indicate the reconfiguration for service k between $t-1$ and t, where $x_{ik}(t-1) \oplus x_{ik}(t) = 1$ if $x_{ik}(t)$ and $x_{ik}(t+1)$ have different values, otherwise, $x_{ik}(t-1) \oplus x_{ik}(t) = 0$.

C. Service Coverage and Robustness Model

The area all over the world is denoted by $\mathcal{A}$. The substellar point of satellite i at time slot t can be denoted by $e_i(r, \theta_i(t), \phi_i(t))$ in which r is the earth radius, $\theta_i(t)$ is $(\pi/2) - \textit{latitude}$, $\phi_i(t)$ is the *longitude*. The point set $\mathcal{A}_i(t)$ of all points within the area covered by satellite i at time slot t can be expressed as

$$\begin{aligned} \mathcal{A}_i(t) = \{e(r, \theta, \phi) |\, &\sin\theta \sin\theta_i(t) \cos(\phi - \phi_i(t)) \\ &+ \cos\theta \cos\theta_i(t) \leq \cos\psi\} \end{aligned} \tag{1}$$

where ψ is the half cone angle of the covered point to the earth's core.

We divide the global area $\mathcal{A}$ into M disjoint small regions, denoted by $\mathcal{A} = \{1, \ldots, j, \ldots, M\}$. The region j can access the service on SEC node i only if it is covered by i. Let $\mu_{ij}(t) = 1$ denote that SEC node i can cover the region j at time slot t, otherwise, $\mu_{ij}(t) = 0$. Since the motion of the satellite is periodic, $\mu_{ij}(t)$ is known as a prior. We assume that each region j is relatively small compared with the satellite coverage, partial region coverage is not considered. The region set covered by satellite i can be expressed as

$$c_i(t) = \{j | \mu_{ij}(t) = 1 \quad \forall j \in \mathcal{A}\}. \tag{2}$$

We denote the service demand of service k in time slot t with a vector $\mathbf{d}_k(t) = (d_{k1}(t), \ldots, d_{kj}(t), \ldots, d_{kM}(t))$, where $d_{kj}(t)$

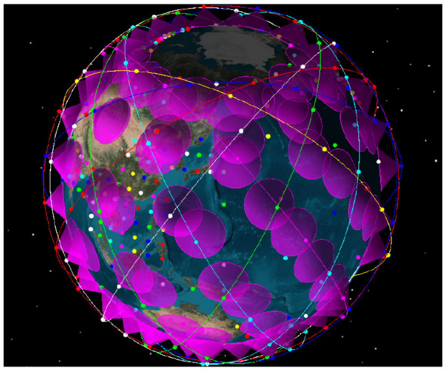


Fig. 3. Coverage areas of the walker constellation.

is the task arrival for service k in region j. In practice, the service demand can be well estimated with high accuracy [44]. Motivated by the radio coverage, the service coverage of service k deployed on satellite i is defined as the number of user requests served by service k considering unbalanced user request distribution. The service coverage of all SEC nodes at time slot t can be expressed as

$$C(\mathbf{x}(t)) = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in c_i(t)} d_{kj}(t)x_{ik}(t). \tag{3}$$

Despite the assumption that ISLs are adopted in constellations, the service access over multiple hops of ISLs is out of the scope of this article.

Given a pair of SEC nodes i and i', we use $\nu_{ii'j}(t) = \mu_{ij}(t)\mu_{i'j}(t)$ to denote if they cover the same region j at time slot t, where $i' \neq i$. Intuitively, $\nu_{ii'j}(t) = \nu_{i'ij}(t)$. And we define $\nu_{iij}(t) = 0$. Two SEC nodes i and i' are neighbors if they cover the same region j, i.e., $\nu_{ii'j}(t) = 1$. The relationship between SEC nodes can be expressed by an undirected graph $\mathbf{G} = (\mathcal{S}, \mathcal{E})$, in which $\mathcal{S}$ is the vertex (SEC nodes) set and the $\mathcal{E}$ is the edge set. An edge exists between two neighboring SEC nodes i and i'. The *service robustness* is defined as the number of user requests co-covered by each service k deployed on both SEC nodes i and i', which can be expressed as

$$R(\mathbf{x}(t)) = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{i'=1}^{N}\sum_{j=1}^{M} w(t)\nu_{ii'j}(t)d_{kj}(t)x_{i'k}(t)x_{ik}(t) \tag{4}$$

where $w(t) \in [0, \infty)$ is the weight constant that is positively related to the preference on the service robustness.

We give a motivating example in Fig. 2 to illustrate that the service coverage and the service robustness are two conflict metrics. From Fig. 2(a), we consider a scenario where three copies of a service can be deployed in three of the five SEC nodes $\{S_1, S_2, S_3, S_4, S_5\}$ and the user requests are assumed to be distributed uniformly. Thus, the service coverage can be evaluated by the union of satellite coverage area while the service robustness is evaluated by the intersection of satellite coverage area.In Fig. 2(b), the service placement decision $\{S_2, S_4, S_5\}$ maximizes the service coverage but minimizes the service robustness. In Fig. 2(c), the service placement decision $\{S_1, S_2, S_3\}$ maximizes the service robustness but minimizes the service coverage. This reflects the conflict between service coverage and service robustness.

D. Problem Formulation

Service coverage focuses on the coverage over local regions with service requests. It generally requires choosing a smaller number of satellites to deploy services to achieve on-demand service coverage. The dynamic service placement in SEC aims to maximize robustness aware service coverage

$$RC(\mathbf{x}(t)) = C(\mathbf{x}(t)) + R(\mathbf{x}(t)). \tag{5}$$

The problem can be formulated as

$$\begin{aligned}
\mathbf{P1}\quad & \max_{\mathbf{x}(1),\ldots,\mathbf{x}(T)} \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1} RC(\mathbf{x}(t))\\
\text{s.t.}\quad & \text{(C1)}\quad \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N} x_{ik}(t) \le B_k \quad \forall k \in \mathcal{F}\\
& \text{(C2)}\quad \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N} x_{ik}(t-1) \oplus x_{ik}(t) \le O_k \quad \forall k \in \mathcal{F}\\
& \text{(C3)}\quad \sum_{k=1}^{K} p_k x_{ik}(t) \le P_i \quad \forall i \in \mathcal{S} \quad \forall t \in \mathcal{T}.
\end{aligned} \tag{6}$$

Here, constraint (C1) ensures the long-term averaged number of service copies deployed at all edge nodes does not exceed the given budget B_k. Constraint (C2) ensures that the long-term averaged reconfiguring times of each service k donot exceed the predefined numbers O_k. Constraint (C3) ensures that the computation demand of services in each SEC node i does not exceed its capacity in each time slot t. To solve problem **P1**, the proposed algorithm needs to make service placement decisions to adapt to system dynamics such as service demand and satellite mobility in each time slot. Because network conditions and user behavior are hard to predict in a long run. Leveraging Lyapunov optimization theory [45], the long-term constraints (C1)-(C2) control the queue stability and long-term problems can be decoupled into real-time ones. This method solves the real-time optimization problem online without requiring future system information.

IV. Algorithm Design

A. Lyapunov Optimization-Based Online Service Placement

We develop an online service placement algorithm under the Lyapunov optimization framework. As the service placement decision is made dynamically, the service placement and operational cost in different time slots can exceed or remain under the budget limit. To characterize the historical exceeded service placement and operational cost, we define virtual queues for each service k as

$$Q_k^B(t+1) = Q_k^B(t) + \max\left\{\sum_{i=1}^{N} x_{ik}(t) - B_k, 0\right\} \tag{7}$$

$$Q_k^O(t+1) = Q_k^O(t) + \max\left\{\sum_{i=1}^{N} x_{ik}(t-1) \oplus x_{ik}(t) - O_k, 0\right\} \tag{8}$$

where $Q_k^B(t)$ and $Q_k^O(t)$ are the queue lengths at time slot t, and initialized as 0 [i.e., $Q_k^B(0) = 0, Q_k^O(0) = 0$] and $x_{ik}(-1) \oplus x_{ik}(0) = 0$. The queues indicate that $Q_k^B(t)$ and $Q_k^O(t)$ accumulate the excessive service copies and operational cost over the budget limit. Constraints (C1) and (C2) can be ensured by enforcing the stability of $\mathbf{Q}(t) = (Q_k^B(t), Q_k^O(t))$ [45]. The quadratic Lyapunov function is defined as

$$\mathcal{L}(\mathbf{Q}(t)) = \frac{1}{2}\sum_{k=1}^{K}\left[\left(Q_k^B(t)\right)^2 + \left(Q_k^O(t)\right)^2\right] \tag{9}$$

and one step conditional Lyapunov drift function is defined as

$$\Delta(\mathbf{Q}(t)) = \mathbb{E}[\mathcal{L}(\mathbf{Q}(t+1)) - \mathcal{L}(\mathbf{Q}(t))|\mathbf{Q}(t)]. \tag{10}$$

It can be inferred from [45] that the smaller $\Delta(\mathbf{Q}(t))$ (for each $t \in \mathcal{T}$) is, the more likely $\mathbf{Q}(t)$ is stabilized. We seek to find a service placement solution that balances the service coverage performance and cost. Lyapunov drift-plus-penalty function $\Delta(\mathbf{Q}(t)) - V\mathbb{E}[RC(\mathbf{x}(t))|\mathbf{Q}(t)]$ is introduced to solve the problem, where V is a nonnegative parameter that controls the tradeoff between service coverage performance and cost queue length. From the definition of $\mathbf{Q}(t)$, we can derive that the drift-plus-penalty is upper bounded as

$$\begin{aligned}\Delta(\mathbf{Q}(t)) - V\mathbb{E}[RC(\mathbf{x}(t))|\mathbf{Q}(t)] &\le \Delta_B \\ + \mathbb{E}\Bigg[\sum_{k=1}^{K} Q_k^B(t)\left(\sum_{i=1}^{N} x_{ik}(t) - B_k\right) &\sum_{k=1}^{K} Q_k^O(t) \\ \left(\sum_{i=1}^{N} x_{ik}(t-1) \oplus x_{ik}(t) - O_k\right)& \\ - V \cdot RC(\mathbf{x}(t))|\mathbf{Q}(t)\Bigg]&\end{aligned} \tag{11}$$

where $\Delta_B = 1/2\sum_{k=1}^{K}[(N-B_k)^2 + (N-O_k)^2]$ for each time slot t. According to the Lyapunov drift-plus-penalty framework in [45], we just need to optimize the bound on the right side of (11) in each time slot t, then the problem **P1** is transformed into

$$\begin{aligned}\mathbf{P2}\text{: min } &\sum_{k=1}^{K}(Q_k^B(t)\sum_{i=1}^{N} x_{ik}(t) + Q_k^O(t) \\ &\sum_{i=1}^{N} x_{ik}(t-1) \oplus x_{ik}(t)) + \sum_{k=1}^{K} p_k x_{ik}(t) - V \cdot RC(\mathbf{x}(t)).\end{aligned} \tag{12}$$

Till now, we have transformed the original dynamic optimization problem **P1** to a static optimization problem **P2** in each time slot. We describe the online service placement algorithm in Algorithm 1. In each time slot t, it first solves the problem **P2** to obtain a near-optimal solution, and then update the placement and operational cost queues.

In the following, we introuduce Markov approximation [46] and parallel Gibbs sampling method [47] to solve problem **P2**.

Algorithm 1 Online Service Placement Algorithm

```
Input:
    Preference factors $w(t)$, service demand $\mathbf{d}_k(t)$, coverage indicator $\mu_{ij}(t)$.
Output:
    Service placement decision $\mathbf{x}^*(t), t = \{1, \cdots, T\}$.
Initialization: $Q_k^B(0) = 0$, $Q_k^O(0) = 0$, $\forall k$.
for Each time slot $t = 1, \cdots, T$ do
    Solve the problem $\mathbf{P2}$:$\mathbf{x}^*(t) = \min(12)$.
    Update the length of cost queues according to (7) and (8) based on $\mathbf{x}^*(t)$.
end for
```

B. Parallel Gibbs Sampling

We design a service placement algorithm that can obtain a near-optimal solution. We first prove the NP-hardness of problem **P2**.

1) Complexity Analysis of ***P2****:* We present the NP-hardness of **P2** by analyzing one simplified case considering only one type of service ($K = 1$) with a given number of service copies [i.e., $\sum_{i=1}^{N} x_{ik}(t) = B$] without operational cost [i.e., $\sum_{i=1}^{N} x_{ik}(t-1) \oplus x_{ik}(t) = 0$] and SEC node capacity [i.e., constraint (C3)]. Thus, the objective of **P2** can be written as $Q_k^B(t)B - V \cdot RC(\mathbf{x}(t))$. The simplified problem **P2-S** can be expressed as

$$\begin{aligned}\mathbf{P2-S}\text{: } &\max \quad RC(\mathbf{x}(t)) \\ &\text{s.t. } \sum_{i=1}^{N} x_{i1}(t) = B.\end{aligned} \tag{13}$$

Problem **P2-S** is NP-hard since it is reducible from the *k-heaviest-subgraph* problem [48]. Since the service placement decisions of SEC nodes are coupled, problem **P2** is complicated. Optimization solvers, e.g., IBM-CPLEX [49], usually solve this kind of problem in a centralized manner, which is computationally prohibitive. Therefore, we propose a decentralized algorithm with low time complexity based on Markov approximation and Gibbs sampling.

2) Algorithm Design: As we perform the optimization in each time slot t, we omit the time index for ease of description in the following part. We denote $U(\mathbf{x})$ as the objective function of problem **P2** in each time slot t, where $\mathbf{x} \in \mathcal{X}$ is the service placement decision. Problem **P2** is transformed into the convex log-sum-exp problem [46]

$$\begin{aligned}\min_{p(\mathbf{x})} \quad &\sum_{\mathbf{x}\in\mathcal{X}(t)} p(\mathbf{x})U(\mathbf{x}) + \frac{1}{\tau}\sum_{\mathbf{x}\in\mathcal{X}(t)} p(\mathbf{x})\log p(\mathbf{x}) \\ \text{s.t.} \quad &\sum_{\mathbf{x}\in\mathcal{X}(t)} p(\mathbf{x}) = 1 \quad \forall t \in \mathcal{T}\end{aligned} \tag{14}$$

where $p(\mathbf{x})$ is the probability of the service placement decision $\mathbf{x}$. τ is the approximation ratio of the entropy term. The problem in (14) becomes the problem **P2** as $\tau \to \infty$. The optimal solution can be obtained as [46]

$$p^*(\mathbf{x}) = \frac{e^{-\tau U(\mathbf{x})}}{\sum_{\mathbf{x}'} e^{-\tau U(\mathbf{x}')}}. \tag{15}$$

According to probability $p^*(\mathbf{x})$, we can get the optimal service placement decision. The basic idea is to constantly update service placement decisions to form a Markov chain, which is irreducible that traverses all feasible states. There are two drawbacks if we update placement decision centralized by randomly altering the placement decision of each SEC node. First, the time complexity is high when the exploration space (determined by the number of SEC nodes and service types) is large. Second, it needs to collect the global information in the whole SEC system, which is hard sometimes.

In this article, we resort to Gibbs sampling to approach the stationary distribution in (15) (also called *Gibbs distribution*), which can be guaranteed by the Markov chain Monte Carlo theory. The sequential Gibbs sampling iteratively samples service placement strategies of different SEC nodes according to the following distribution:

$$\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_{-i}) = \frac{e^{-\tau \sum_{n \in \mathcal{S}} U_n(\mathbf{x}_i, \mathbf{x}_{-i})}}{\sum_{\mathbf{x}'_i \in \mathcal{X}_i} e^{-\tau \sum_{n \in \mathcal{S}} U_n(\mathbf{x}'_i, \mathbf{x}_{-i})}} \tag{16}$$

where $\mathcal{X}_i(t)$ is the feasible decision set, $\mathbf{x}_i \in \mathcal{X}_i$ is the service placement decision of SEC node $i \in \mathcal{S}$ and $\mathbf{x}_{-i}$ represent the joint decision of all SEC nodes except i. We also use n to denote an SEC node to distinguish from SEC node i, U_n is the n_{th} part of the objective function in (12) and $U(\mathbf{x}) = \sum_{n \in \mathcal{S}} U_n(\mathbf{x})$. The joint posterior distribution guarantees that Gibbs sampling converges to the Gibbs distribution. However, the sequential Gibbs sampling has the same drawbacks as centralized updating. One intuitive idea is to introduce parallelism into the sequential Gibbs sampling. If the decision update of SEC node i do not influence the sampling distribution $p(\mathbf{x}_{i'}, \mathbf{x}_{-i'})$ of SEC node i', then SEC nodes i and i' can make decisions simultaneously. Markov Random Field theory can support this intuition. It is an undirected graphical model in which each node corresponds to the service placement decision variable and the edges identify the neighbor relationship between nodes. Markov blanket of SEC node i is the set of SEC nodes adjacent to SEC node i denoted by Γ_i. The service placement decision of SEC node i is conditionally independent of that of the SEC nodes not in the Γ_i, that is, $p(\mathbf{x}_i|\mathbf{x}_{\Gamma_i}) = p(\mathbf{x}_i|\mathbf{x}_{-i})$.

As we mentioned that two SEC nodes i and i' are neighbors if there exists any region j such that $v_{ii'j}(t) = 1$. We define the set of the neighbor of i on graph $\mathbf{G}$ as one-hop neighbors denoted by $\Omega_{i,1}$ and the neighbors of the one-hop neighbors of i (excluding itself) as two-hop neighbors denoted by $\Omega_{i,2}$. In the following proposition, we will show the mapping of Markov Random Field on our physical network $\mathbf{G}$ defined in the system model.

Proposition 1: The Markov blanket of SEC node i on the Markov Random Field is the set of SEC nodes in $\Omega_{i,1}$ and $\Omega_{i,2}$ on physical graph $\mathbf{G}$, namely, $\Gamma_i = \Omega_{i,1} \cup \Omega_{i,2}$. Moreover, the sampling probability can be rewritten as

$$\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_{-i}) = \frac{e^{-\tau \sum_{n \in \Omega_{i,1}} U_n(\mathbf{x}_i, \mathbf{x}_{\Gamma_i})}}{\sum_{\mathbf{x}'_i \in \mathcal{X}_i} e^{-\tau \sum_{n \in \Omega_{i,1}} U_n(\mathbf{x}'_i, \mathbf{x}_{\Gamma_i})}}. \tag{17}$$

Proof: See the proof of [47, Proposition 1]. ■

Proposition 1 implies that SEC nodes more than two hops away from i has no influence on its sampling probability. This is critical to enable parallel sampling of SEC nodes' decisions. We divide SEC nodes into $L(L \leq N)$ different groups so that all the SEC nodes in the same group can update their placement decision simultaneously. We seek to minimize L to maximize the parallelization level, which can be transformed into a graph coloring problem on the Markov Random Field. We can solve it by a sequential coloring algorithm [47]. Let $\mathcal{S}_l$ be the set of SEC nodes with color l. The proposed algorithm works iteratively. It first chooses an SEC node set $\mathcal{S}_l$. When SEC nodes in $\mathcal{S}_l$ make service placement decisions, they need the service demand patterns of one-hop neighbors and the service placement decision in the Markov blanket. Based on this information, SEC node i can compute $\sum_{n \in \Omega_{i,1}} U_n(\mathbf{x}_i, \mathbf{x}_{\Gamma_i})$ for each service placement decision $\mathbf{x}_i$, then SEC node i update to a new decision according to the probability distribution in (17). To save communication overhead, SEC nodes can change the service placement decision when Algorithm 2 converges to the global optimal solution. As we mentioned, τ is the approximation ratio of the convex log-sum-exp problem. From (17), we find that τ influences the decision updating probability. When $\tau \to \infty$, the algorithm becomes greedy. If a decision has a smaller value of $\sum_{n \in \Omega_{i,1}} U_n(\mathbf{x}_i, \mathbf{x}_{\Gamma_i})$, the algorithm keeps it with a greater probability. When τ is getting smaller, the gap between probabilities of different decisions will be reduced, the algorithm tries to explore all possible decisions without convergence. To balance the exploration and exploitation, τ is changed adaptively: τ is small at the beginning to explore all the decisions and increases over iterations. The algorithm is stated in Algorithm 2.

3) Time Complexity Analysis of Algorithm 2: The complexity of Algorithm 2 is dominated by triple for-loops. We denote I as the iterations needed in the outer loop to convergence. In the middle loop, there is L iterations as we use graph coloring to generate L SEC node sets. In the inner loop, the iteration number is determined by the decision space size of each SEC node, which is smaller than $\max_{i \in \mathcal{S}} |\mathcal{X}_i|$. The complexity of each operation in the inner loop is $O(1)$. Therefore, the complexity of Algorithm 2 can be quantified by $O(I \cdot L \cdot \max_{i \in \mathcal{S}} |\mathcal{X}_i|)$.

Remarks 1: As Algorithm 2 is part of Algorithm 1, two input information need to be calculated in Algorithm 1 based on global information, one is physical network topology, the other is the queue length. The global information can be calculated in any authorized SEC node, and broadcast to all SEC nodes. The physical network topology can be obtained because the satellite motion is predictable. When the joint service placement decisions of all SEC converges to the global optimum in each time slot, they need to transmit the service placement decisions over their ISLs to the SEC node calculating the global information to update each queue length.

C. Algorithm Performance Analysis

In this part, we analyze theoretically our online service placement algorithm for SEC. First, we prove the convergence of Algorithm 2. Then we discuss the optimality gap

Algorithm 2 Service Placement Based on Gibbs Sampling

```
Input:
    Preference factors $w(t)$, service demand $\mathbf{d}_k(t)$, coverage indicator $\mu_{ij}(t)$, physical network graph $\mathbf{G}$, queue length $Q_k^B(t)$, $Q_k^O(t)$, approximation ratio $\tau$.
Output:
    Service placement decision $\mathbf{x}^*(t)$.
Construct the Markov random field based on $\mathbf{G}$ and $L$ coloring sets.
for Each iteration iter do
    Pick a SEC node set $\mathcal{S}_l$ according to a predefined order;
    for Each SEC node $i$ in $\mathcal{S}_l$ do
        for Each feasible service placement decision $\mathbf{x}_i$ in $\mathcal{X}_i(t)$ do
            Calculate the value of $\sum_{n\in\Omega_{i,1}} U_n(\mathbf{x}_i, \mathbf{x}_{\Gamma_i})$.
        end for
        Set $\mathbf{x}_i^*(iter+1) = \mathbf{x}_i$ according to the probability in (17) and send the decision to all SEC nodes in $\Gamma_i$.
    end for
    Each SEC node $i \notin \mathcal{S}_l$, keep $\mathbf{x}_i^*(iter+1) = \mathbf{x}_i^*(iter)$.
    Stop iterations when the value of objective function in (12) converges.
end for
Return the service placement decision $\mathbf{x}^*$.
```

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Total decision round number, T	180
Service type number, K	5
SEC node number, N	144
Ground region number, M	354
Average service demand arriving rate, d_{kj}	(0, 10)
SEC node computation capacity, P_i	[1, 2] Giga cycles
Service computation requirement p_k, β_s	[0.5, 1] Giga cycles/service
Service placement budget, B_k	25 (services)
Service adjustment budget, O_k	5 (services)
Approximation ratio parameter, τ	500
Lyapunov control paramenter, V	100
Service robustness preference parameter, $w(t)$	[0, 10]
Each decision round duration	8 minutes

of Algorithm 2. Finally, we compare Algorithm 1 with the offline optimum.

Theorem 1: Algorithm 2 converges from any starting state of the Markov chain to Gibbs distribution $p^*(\mathbf{x})$ in (15).

Proof: See the proof of [47, Proposition 2]. ■

As stated in Theorem 1, Algorithm 2 can achieve the optimal probability distribution for the convex log-sum-exp problem in (14). We denote the minimum supremum bound and expected supremum bound derived by Algorithm 2 as $U^*(\mathbf{x})$ and $\tilde{U}(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{X}(t)} p^*(\mathbf{x})U(\mathbf{x})$, respectively.

Theorem 2: The optimality gap of Algorithm 2 is given as

$$\tilde{U}(\mathbf{x}) - U^*(\mathbf{x}) \leq \frac{1}{\tau}\log|\mathcal{X}| \tag{18}$$

where $|\mathcal{X}|$ is the number of feasible decisions of all SEC nodes.

Proof: See the proof of [46, Th. 1]. ■

Theorem 2 implies that Algorithm 2 can achieve minimum supremum bound $U^*(\mathbf{x}))$ when $\tau \to \infty$. The following theorem demonstrates the suboptimality of Algorithm 1 compared with the offline optimum.

Theorem 3: Following the optimal online service placement decision $\mathbf{x}^*(t)$ obtained by Algorithm 1, the long-term robustness aware service coverage satisfies

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[RC(\mathbf{x}(t))] \geq RC^* - \frac{\Delta_B}{V} - \frac{\log|\mathcal{X}|}{\tau V} \tag{19}$$

and the long-term cost deficit satisfies

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{k=1}^{K}\left\{\mathbb{E}\left[\sum_{i=1}^{N}x_{ik}(t) - B_k\right] + \mathbb{E}\left[\sum_{i=1}^{N}x_{ik}(t-1)\oplus x_{ik}(t) - O_k\right]\right\} \leq \frac{\Delta_B + V\big(RC^* - RC^{\min}\big)}{\epsilon} + \frac{1}{\epsilon\tau}\log|\mathcal{X}| \tag{20}$$

where RC^* is the optimal robustness aware service coverage to **P1**, $RC^{\min}$ is the smallest robustness aware service coverage, and $\epsilon > 0$ is a constant.

Proof: See the Appendix. ■

In Theorem 3, our algorithm can achieve a lower bound of the time-averaged service coverage in (19) and an upper bound for the sum of the service placement and operational cost deficit in (20). For any parameter $V > 0$, the time averaged service coverage differs from the optimal value RC^* by no more than $(\Delta_B/V) - [(\log|\mathcal{X}|)/(\tau V)]$, which can be made arbitrarily small as V is increased. However, the time average cost deficit bound increases linearly with the parameter V, as shown by (20). This presents a performance-cost tradeoff of $[O(1/V), O(V)]$ in our online algorithm, where we can set the parameter V to a desirable value to achieve the balance of the long-term service coverage performance and the cost.

V. SIMULATION RESULTS

A. Simulation Settings

In this section, we conduct extensive simulations to evaluate our algorithm. We simulate a Walker-Delta constellation in AGI's STK software according to the rules in [50]. As shown in Fig. 3, the constellation has 12 orbital planes, each of which consists of 12 satellites at an altitude of 829 km with an inclination of 60°. Each satellite has a communication coverage (pink circles) and different satellites have overlapped coverage. Coverage at a location varies over time as satellites move in and out of view. To simulate the service demand distribution on the earth's surface, we divide the earth's surface into 354 regions by latitude and longitude with different areas according to global population distribution data [51]. About 75% of regions are on the ground while about 25% regions on the sea [51]. Fig. 4 shows the distribution of regions. We set the service type number as 5. Each region has simulated service requests generated by a ground station in the center. The service demand of each region is modeled as a Poisson process

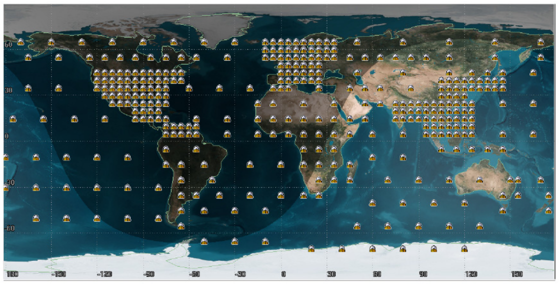


Fig. 4. Ground region division illustration.

with an arrival rate randomly generated from 0 to 10 [47]. And we assume that the service demand of each region is generated in the center of the region. We use AGI's STK to simulate the satellite motion for one day and collect the connection data between each satellite and each region every one second. We make service placement decisions every eight minutes because the service placement decision cannot change very frequently while each sate-ground link session can last between a few seconds and ten minutes [19]. So we have total $180(= 60*24/8)$ decision rounds. The computing capacity of SEC nodes is randomly set as [1, 2] Giga CPU cycles and each service need a computation capacity of [0.5, 1] Giga CPU cycles. The placement budget of each service is set as 25 (services) and the operational cost of each service is set as 5 (services). The approximation ratio τ is set as 500. The service robustness preference parameter is randomly set as [0, 10]. We compare our algorithm under different approximation ratio parameters ($\tau = 500$, $\tau = 30*$iteration) with a random service placement method labeled as *Random*. In each decision round, SEC nodes choose a feasible service placement decision only considering the service placement budget. The main parameters are shown as Tabel I. Next, we evaluate our algorithm under different parameter settings.

B. Performance Comparison With Time

First, we evaluate the performance and average cost of our algorithm along with time slots. We compare our algorithm under different approximation ratio parameters with the Random algorithm. From Fig. 5(a), we can observe that our algorithms achieve average 4.3× and 3.9× robustness aware service coverage of Random algorithm when $\tau = 30*iteration$ and $\tau = 500$, respectively. The coverage performance fluctuates over different time slots because the service demand of each time slot is randomly generated. We can also observe that the robustness service coverage is higher when τ is changed adaptively, i.e., $\tau = 30 * iteration$ and the reason will be explained in the convergence analysis of Fig. 7(a). In Fig. 5(b), we can observe that the service placement cost of all three algorithms converges to 25. The adjustment cost of our algorithm converges to 5 while that of the Random algorithm converges to 35 which violates the operational cost budget. The reason is that the Random algorithm randomly makes service placement decisions without considering operational cost budget.

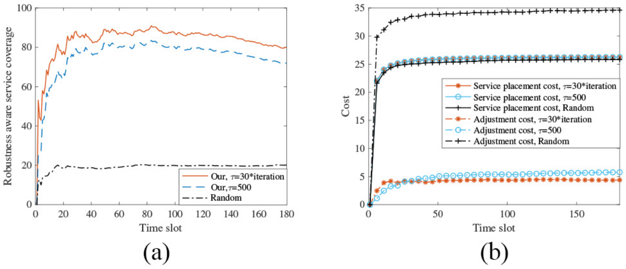


Fig. 5. Performance comparison along with time. (a) Service coverage variation. (b) Cost variation.

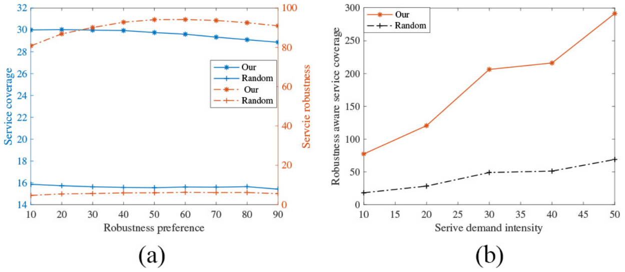


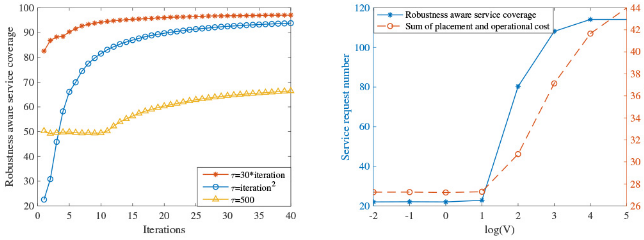
Fig. 6. Performance comparison under different parameters. (a) Robustness preference impact. (b) Service demand intensity impact.

C. Performance Comparison With Different Parameters

Next, we evaluate the algorithm performance under different robustness preference settings and service demand intensity settings. In these two evaluations, we set $\tau = 30 * iteration$ and compare with Random algorithm. In the robustness preference impact evaluation, we vary robustness preference in the range of [10, 90] and record the value of service coverage and service robustness, respectively, in Fig. 6(a). As expected, the service coverage decrease because it is less important when the robustness preference increases. The service robustness first increases and then keeps unchanged with the robustness preference because the service robustness is more important and increases to the maximal value which cannot increase anymore. Both the service coverage and service robustness of Random algorithm keep unchanged with the robustness preference because the Random algorithm randomly makes service placement decisions without considering the objective function. In the service demand intensity impact evaluation, we vary the average service demand arrival rate in the range of [10, 50] and record the value of robustness aware service coverage. From Fig. 6(b), our algorithm achieves on average 4.3× robustness aware service coverage of Random algorithm and the robustness aware service coverage of both algorithms increases with the service demand intensity as we expected. However, the increasing trend of both algorithms is not strictly linear. Since this result is not algorithm-specific, we think that it is caused by service request arrival randomness. So we can infer that our algorithm has good scalability with service demand intensity.

D. Algorithm Parameter Analysis

In the following, we verify the theoretical result of convergence in Theorem 1 by simulations in Fig. 7(a). We have



(a) (b)

Fig. 7. Algorithm parameter analysis. (a) Convergence under different approximation ratio parameters. (b) Control parameter impact.

analyzed in the algorithm design part that Algorithm 2 converges to the global optimal solution when $\tau \rightarrow \infty$ and different values lead to different convergence performance. Hence, we evaluate Algorithm 2 under three typical settings of τ, i.e., a constant value $\tau = 500$, a linear function of iterations $\tau = 30 * iiteration$, and a quadratic function $\tau = iteration^2$. From Fig. 7(a), we can observe that our algorithm has the best performance when $\tau = 30 * iteration$ while our algorithm has the worst performance when $\tau = 500$. As we mentioned that τ influences the decision updating probability. When $\tau \rightarrow \infty$, our algorithm can be stuck in a local optimum because it becomes greedy and keeps an updated decision with a greater probability if it incurs a smaller value of $\sum_{n\in\Omega_{i,1}} U_n(\mathbf{x}_i, \mathbf{x}_{\Gamma_i})$. When τ is getting smaller, the gap between probabilities of different decisions will be reduced, our algorithm tries to explore all possible decisions without convergence. So when τ is increased from a small value adaptively, the algorithm performance is better. We also find that the linear function has better performance than higher power functions because higher power function makes τ to increase greatly and leads to a local optimum.

Fig. 7(b) shows the impact of Lyapunov parameter V on robustness aware service coverage and the sum of service placement and operational cost of each service. By increasing V from 10^{-2} to 10^5, our algorithm cares more about the service coverage performance and thus the robustness service coverage increases to a maximal value and keeps unchanged. However, with less concern on the service placement and operational cost, the cost increases. The coverage-cost performance follows the $[O(1/V), O(V)]$ tradeoff as stated in Theorem 3. At the same time, the results also provide an insight for selecting V in practice.

VI. Conclusion

In this article, we investigate dynamic service placement in SEC to achieve robustness aware service coverage with a limited budget. We formulate the problem as a stochastic optimization problem with long-term averaged objective function and constraints. Then, we propose a novel online algorithm to transform the long-term averaged problem into real-time optimization problems in each time slot and solve them online leveraging Lyapunov optimization theory, Markov approximation method, and Gibbs sampling algorithm. We prove that our algorithm can converge to a near-optimal result and the optimality gap is with a theoretical bound. We carry out extensive simulations to evaluate our algorithm and the simulation results show that our algorithm outperforms the baseline. In the future work, we will focus on cooperative computation by allocating communication and computing resources for space service computing to improve users' quality of experience such as latency in various traffic conditions.

Appendix
Proof of Theorem 3

We first introduce the following Lemma to prove the time-averaged service coverage bound.

Lemma 1: For an arbitrary $\delta > 0$, a stationary and randomized policy $\bar{\mathbf{x}}(t)$ for **P2** exists to make service placement decisions independent of the queue state such that

$$\begin{aligned} &\mathbb{E}[RC(\bar{\mathbf{x}}(t))] \geq RC^* - \delta \\ &\mathbb{E}\left[\sum_{i=1}^{N} \bar{x}_{ki}(t) - B_k\right] \leq \delta \quad \forall k \\ &\mathbb{E}\left[\sum_{i=1}^{N} \bar{x}_{ki}(t-1) \oplus \bar{x}_{ki}(t) - O_k\right] \leq \delta \quad \forall k. \end{aligned} \tag{21}$$

Proof: See the proof of [45, Th. 4.5]. ■

Algorithm 1 seeks to choose strategies $\mathbf{x}^*$ to minimize **P2**. Based on the Theorem 2, the optimality gap is $(1/\tau)(\log|\mathcal{X}|)$ over all time slots. By applying Lemma 1 into the drift-plus-penalty inequality, we obtain

$$\begin{aligned} &\Delta(\mathbf{Q}(t)) - V\mathbb{E}\big[RC\big(\mathbf{x}^*(t)\big)|\mathbf{Q}(t)\big] \\ &\leq \Delta_B + \frac{1}{\tau}\log|\mathcal{X}| + \mathbb{E}\Bigg[Q_k^B(t)\left(\sum_{i=1}^{N}\bar{x}_{ki}(t) - B_k\right) \\ &\qquad + Q_k^O(t)\left(\sum_{i=1}^{N}\bar{x}_{ki}(t-1)\oplus\bar{x}_{ki}(t) - O_k\right) \\ &\qquad - V\cdot RC(\bar{\mathbf{x}}(t))|\mathbf{Q}(t)\Bigg] \\ &\overset{(a)}{\leq} \Delta_B + \frac{1}{\tau}\log|\mathcal{X}| + Q_k^B(t)\delta + Q_k^O(t)\delta - V\cdot(RC^* + \delta). \end{aligned} \tag{22}$$

The inequality (a) is because the policy $\bar{\mathbf{x}}$ is independent of the cost deficit queue. Let δ go to zero, sum the inequality over $t \in 0, 1, \ldots, T-1$ and then divide the result by T, we have

$$\begin{aligned} &\frac{1}{T}\mathbb{E}[\mathcal{L}(\mathbf{Q}(T-1)) - \mathcal{L}(\mathbf{Q}(0))] - \frac{V}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[RC\big(\mathbf{x}^*(t)\big)\big] \\ &\leq \Delta_B + \frac{1}{\tau}\log|\mathcal{X}| - V\cdot RC^*. \end{aligned} \tag{23}$$

Considering that $\mathcal{L}(\mathbf{Q}(T-1)) \geq 0$ and $\mathcal{L}(\mathbf{Q}(0)) = 0$, we have the bound for long-term service coverage.

To obtain the long-term cost deficit bound, we assume that there are a value ϵ and a policy $\tilde{\mathbf{x}}$ that satisfies

$$\mathbb{E}\left[\sum_{i=1}^{N}\tilde{x}_{ki}(t) - B_k\right] \leq -\epsilon \quad \forall k \tag{24}$$

$$\mathbb{E}\left[\sum_{i=1}^{N}\tilde{x}_{ki}(t-1)\oplus\bar{x}_{ki}(t)-O_k\right]\le-\epsilon\quad\forall k. \tag{25}$$

Plugging above into (11), we have

$$\begin{aligned}\Delta(\mathbf{Q}(t))-V\cdot\mathbb{E}\big[RC\big(\mathbf{x}^*(t)\big)|\mathbf{Q}(t)\big]&\le\Delta_B+\frac{1}{\tau}\log|\mathcal{X}|\\ &-V\cdot RC\left(\tilde{\mathbf{x}}(t)-\epsilon\left(\sum_{k=1}^{K}Q_k^B(t)+\sum_{k=1}^{K}Q_k^O(t)\right)\right).\end{aligned} \tag{26}$$

Summing the above over $t\in\{0,1,\dots,T-1\}$, we have

$$\begin{aligned}&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{k=1}^{K}\mathbb{E}\Big[Q_k^B(t)+Q_k^O(t)\Big]\\ &\le\frac{1}{\epsilon}\Delta_B+\frac{1}{\epsilon\tau}\log|\mathcal{X}|+\frac{V}{\epsilon T}\sum_{t=0}^{T-1}\Big(RC\big(\mathbf{x}^*(t)\big)-RC\big(\tilde{\mathbf{x}}(t)\big)\Big)\\ &\le\frac{\Delta_B+V\cdot\big(RC(\mathbf{x}^*(t))-RC\big(\tilde{\mathbf{x}}(t)\big)\big)}{\epsilon}+\frac{1}{\epsilon\tau}\log|\mathcal{X}|\\ &\le\frac{\Delta_B+V\cdot\big(RC^*-RC^{\min}\big)}{\epsilon}+\frac{1}{\epsilon\tau}\log|\mathcal{X}|.\end{aligned} \tag{27}$$

Considering $\sum_{t=0}^{T-1}\sum_{k=1}^{K}\mathbb{E}[Q_k^B(t)+Q_k^O(t)]\ge\sum_{t=0}^{T-1}\sum_{k=1}^{K}\{\mathbb{E}[\sum_{i=1}^{N}x_{ik}(t)-B_k]+\mathbb{E}[\sum_{i=1}^{N}x_{ik}(t-1)\oplus\bar{x}_{ki}(t)-O_k]\}$, we have the long-term cost deficit bound.

References

[1] IoT Analytics. (2020). *State of the IoT 2020: 12 Billion IoT Connections, Surpassing Non-IoT for the First Time*. [Online]. Available: https://iot-analytics.com

[2] Y. Wang, J. Yang, X. Guo, and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing," *IEEE Access*, vol. 8, pp. 12510–12520, 2020.

[3] Federal Communications Commision. (2018). *FCC Authorizes SpaceX to Provide Broadband Satellite Services*. [Online]. Available: https://docs.fcc.gov/public/ attachments/DOC-349998A1.pdf

[4] Federal Communications Commision. (2017). *FCC Grants OneWeb US Access for Broadband Satellite Constellation*. [Online]. Available: https://docs.fcc.gov/public/ attachments/DOC-345467A1.pdf

[5] Federal Communications Commision. (2017). *Space Norway NGSO Market Access Grant*. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-17-146A1.pdf

[6] Federal Communications Commision. (2017). *Telesat NGSO Market Access Grant*. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-17-147A1.pdf

[7] L. Boero, R. Bruschi, F. Davoli, M. Marchese, and F. Patrone, "Satellite networking integration in the 5G ecosystem: Research trends and open challenges," *IEEE Netw.*, vol. 32, no. 5, pp. 9–15, Sep./Oct. 2018.

[8] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli, "Integration of satellites in 5G through LEO constellations," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[9] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.

[10] H. Chao, D. E. Comer, and O. Kao, "Space and terrestrial integrated networks: Emerging research advances, prospects, and challenges," *IEEE Netw.*, vol. 33, no. 1, pp. 6–7, Jan./Feb. 2019.

[11] G. Giambene, S. Kota, and P. Pillai, "Satellite-5G integration: A network perspective," *IEEE Netw.*, vol. 32, no. 5, pp. 25–31, Sep./Oct. 2018.

[12] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.

[13] Y. Shi, Y. Cao, J. Liu, and N. Kato, "A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks," *IEEE Netw.*, vol. 33, no. 1, pp. 29–35, Jan./Feb. 2019.

[14] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.

[15] F. Tang, "Dynamically adaptive cooperation transmission among satellite-ground integrated networks," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1559–1568.

[16] A. Varasteh *et al.*, "Mobility-aware joint service placement and routing in space-air-ground integrated networks," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[17] G. Wang, S. Zhou, S. Zhang, Z. Niu, and X. Shen, "SFC-based service provisioning for reconfigurable space-air-ground integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1478–1489, Jul. 2020.

[18] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit computing: An outlandish thought experiment?" in *Proc. ACM Workshop Hot Topics Netw.*, 2020, pp. 197–204.

[19] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2020, pp. 939–954.

[20] L. Yan *et al.*, "SatEC: A 5G satellite edge computing framework based on micro-service architecture," *Sensors*, vol. 19, no. 4, p. 831, 2019.

[21] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Netw.*, vol. 34, no. 3, pp. 224–231, May/Jun. 2020.

[22] Q. Li, S. Wang, A. Zhou, X. Ma, F. Yang, and A. X. Liu, "QoS driven task offloading with statistical guarantee in mobile edge computing," *IEEE Trans. Mobile Comput.*, early access, Jun. 28, 2020, doi: 10.1109/TMC.2020.3004225.

[23] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing," in *Proc. IEEE Int. Conf. Commun. Syst.*, 2018, pp. 450–455.

[24] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.

[25] N. Cheng *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[26] F. A. Salaht, F. Desprez, and A. Lebre, "An overview of service placement problem in fog and edge computing," *ACM Comput. Surveys*, vol. 53, no. 3, p. 65, 2020.

[27] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, "Quality of experience (QoE)-aware placement of applications in fog computing environments," *J. Parallel Distrib. Comput.*, vol. 132, pp. 190–203, Oct. 2019.

[28] H. Tan, Z. Han, X. Li, and F. C. M. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[29] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[30] F. Faticanti, F. De Pellegrini, D. Siracusa, D. Santoro, and S. Cretti, "Cutting throughput with the edge: App-aware placement in fog computing," in *Proc. IEEE Int. Conf. Cyber Security Cloud Comput.*, 2019, pp. 196–203.

[31] V. Souza *et al.*, "Towards a proper service placement in combined fog-to-cloud (F2C)architectures," *Future Gener. Comput. Syst.*, vol. 87, pp. 1–15, Oct. 2018.

[32] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, "BEGIN: Big data enabled energy-efficient vehicular edge computing," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 82–89, Dec. 2018.

[33] M. Z. Khan, S. Harous, S. U. Hassan, M. U. G. Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019.

[34] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. S. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Mar. 5, 2019, doi: 10.1109/TCC.2019.2903240.

[35] D. Bhattacherjee *et al.*, "Gearing up for the 21st century space race," in *Proc. ACM Workshop Hot Topics Netw.*, 2018, pp. 113–119.

[36] M. Handley, "Delay is not an option: Low latency routing in space," in *Proc. ACM Workshop Hot Topics Netw.*, 2018, pp. 85–91.

[37] T. Klenze, G. Giuliari, C. Pappas, A. Perrig, and D. Basin, "Networking in heaven as on earth," in *Proc. ACM Workshop Hot Topics Netw.*, 2018, pp. 22–28.

[38] SpaceX FCC update. (2018). *SpaceXnon-Geostationary Satellite System*. [Online]. Available: https://www.ic.gc.ca/eic/site/smt-gst.nsf/vwapj/SLPB-005-18-SpaceX-attachment2.pdf

[39] Mynaric. (2019). *Annual Report 2019*. [Online]. Available: https://mynaric.com/wp-content/uploads/2020/05/AR-2019-EN-Website.pdf

[40] Spacewatch Global. (2020). *SpaceX Receives FCC Approval to Deploy One Million Starlink 'UFO' User Terminals in US*. [Online]. Available: https://spacewatch.global/2020/03/spacexreceives-fcc-approval-to-deploy-one-million-starlink/

[41] (2018). *Kuiper USASAT-NGSO-8A ITU Filing*. [Online]. Available: https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/8716

[42] SpaceX FCC Filing. (2017). *SpaceX V-Band Non-Geostationary Satellite*. [Online]. Available: https://fcc.report/IBFS/SAT-LOA-20170301-00027/1190019.pdf

[43] J. Fraire, S. Céspedes, and N. Accettura, "Direct-To-Satellite IoT—A survey of the state of the art and future research perspectives," in *Ad-Hoc, Mobile, and Wireless Networks*. Luxembourg City, Luxembourg: Springer, 2019, pp. 241–258.

[44] Z. Liu *et al.*, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. ACM SIGMETRICS/Perform. Joint Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 175–186.

[45] M. Neely, "Stochastic network optimization with application to communication and queueing systems," in *Synthesis Lectures on Communication Networks*, vol. 3. San Rafael, CA, USA: Morgan Claypool Publ., 2010, pp. 1–211.

[46] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6301–6327, Oct. 2013.

[47] L. Chen, C. Shen, P. Zhou, and J. Xu, "Collaborative service placement for edge computing in dense small cell networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 377–390, Feb. 2021.

[48] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between user coverage and network robustness for edge server placement," *IEEE Trans. Cloud Comput.*, early access, Jul. 10, 2020, doi: 10.1109/TCC.2020.3008440.

[49] *IBM CPLEX Optimizer*. Accessed: May 2021. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[50] Astrome. (2015). *The Art of Satellite Constellation Design: What You Need to Know*. [Online]. Available: https://astrome.net/blogs/the-art-of-satellite-constellation-design-what-you-need-to-know/

[51] LandScan Dataset. (2019). *LandScan 2019*. [Online]. Available: https://landscan.ornl.gov/downloads/2019

Qing Li received the B.S. degree in communication engineering from Hebei University, Baoding, China, in 2014, and the M.S. degree in communication engineering from Xidian University, Xi'an, China, in 2017, respectively. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China.

Her research interests include cloud computing and mobile-edge computing.

Shangguang Wang (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

He is currently a Professor with the School of Computing, Beijing University of Posts and Telecommunications, where he is a Vice-Director of the State Key Laboratory of Networking and Switching Technology. He has published more than 150 papers, and his research interests include service computing, cloud computing, and mobile-edge computing.

Prof. Wang served as a General Chair or a TPC Chair of IEEE EDGE 2020, IEEE CLOUD 2020, IEEE SAGC 2020, IEEE EDGE 2018, and IEEE ICFCE 2017, and the Vice-Chair of IEEE Technical Committee on Services Computing from 2015 to 2018. He has been serving as an Executive Vice-Chair of IEEE Technical Committee on Services Computing since 2021, and IEEE Technical Committee on Cloud Computing since 2020.

Xiao Ma received the B.S. degree in telecommunication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2013, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2018.

She is currently a Postdoctoral Fellow with the State Key Laboratory of Networking and Switching Technology, BUPT. From October 2016 to April 2017, she visited the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests include task scheduling, resource deployment and allocation in mobile cloud computing, and mobile-edge computing.

Qibo Sun received the Ph.D. degree in communication and electronic system from Beijing University of Posts and Telecommunication, Beijing, China, in 2002.

He is currently an Associate Professor with the Beijing University of Posts and Telecommunications. His research interests include services computing, the Internet of Things, and network security.

Dr. Sun is a member of the China Computer Federation.

Houpeng Wang received the bachelor's degree in information security from Beijing University of Posts and Telecommunications, Beijing, China, in 2019. He is currently pursuing the master's degree with the University of Chinese Academy of Sciences, Beijing, and Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences.

His research interests include distributed and parallel computing, edge computing, and computer vision.

Suzhi Cao received the bachelor's and master's degree from Tianjin University, Tianjin, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Academy of Opto-Electronics, Chinese Academy of Sciences, Beijing, China, in 2020.

She is currently an Associate Researcher with the Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences. Her research interests include satellite network, edge computing and distributed computing. She also has a strong research interest in spatial information network technology, spatial optical switching, and routing technology.

Fangchun Yang (Senior Member, IEEE) received the Ph.D. degree in communications and electronic systems from Beijing University of Posts and Telecommunications, Beijing, China, in 1990.

He is currently a Professor with Beijing University of Posts and Telecommunications. He has published eight books and more than 100 papers. His current research interests include network intelligence, service computing, and machine games.

Prof. Yang is a fellow of the IET.