Gateway Placement Optimization in LEO Satellite Networks Based on Traffic Estimation

Jianming Guo, David Rincón, *Member, IEEE,* Sebastià Sallent, *Member, IEEE,* Lei Yang, Xiaoqian Chen, and Xianqi Chen

***Abstract*—Using satellite constellations to provide global Internet access services has recently drawn increasing attention. A low-Earth orbit (LEO) satellite network with multiple satellites provides global coverage, low latency, and operates independently, by which it effectively complements terrestrial IP networks. Satellite gateways are located on the ground and can serve as data exchange points between satellite networks and the Internet. As the placement scheme can affect network performance, finding appropriate sites for gateways constitutes a fundamental problem. This paper proposes a gateway placement optimization (GPO) method for LEO satellite networks in order to solve this problem by modeling it as a combination optimization problem. We aim to identify the best gateway locations that can balance traffic loads while using as few gateways as possible. The constraints to be satisfied concern the physical links between gateways and satellites: specifically, link interference, satellite bandwidth, and number of satellite antennas. We use a gravity model to estimate the traffic matrix from/to gateways and satellites, then we adopt and modify the discrete particle swarm optimization (PSO) algorithm to solve this problem. Finally, we apply the GPO method to numerical tests on real satellite constellations. The results indicate that our method performs well and effectively.**

***Index Terms*—Gateway placement, satellite network, low-Earth orbit, traffic estimation, discrete particle swarm optimization.**

I. Introduction

DUE to the low latency, low cost, and high flexibility, low-Earth orbit (LEO) satellites for communication networks are drawing more and more attention compared with their traditional geostationary Earth orbit (GEO) counterparts. Several LEO constellation network projects with hundreds or even thousands of satellites are currently being proposed by commercial organizations [1] such as OneWeb, SpaceX, and Telesat, to name just a few. Some of these have even been partially deployed on orbit. On the other hand, a satellite can provide greater coverage and higher throughput than terrestrial fiber-optic and mobile network infrastructures, which is why dramatic developments in satellite communication networks have occurred in recent years. Furthermore, because new network applications such as the Internet of Things (IoT) are increasingly emerging, terrestrial networks like the current 5G technology cannot exclusively fulfill these multiple requirements. Therefore, researchers are commonly seeking ways to merge the space and terrestrial networks [2] by exploring several issues regarding network architecture [3], resource management [4], and routing algorithms [5], among others. Additionally, new methods are being proposed to solve problems such as satellite load balance [6] and network management [7] by means of machine learning [8] and software-defined network (SDN). Nevertheless, these have not been completely solved due to the intrinsic characteristics of satellite networks, such as the dynamic topology. Other challenges persist, thus requiring further investigation. Among them stands the important gateway placement problem, which has been rarely addressed in previous studies.

As integrated into terrestrial networks, satellite networks provide Internet services that necessarily require a ground infrastructure, i.e., the gateway, to connect to the Internet backbone (data centers, servers, etc.). That is to say, gateways act as edge routers between satellite and terrestrial devices, forming part of the ground segment in the satellite network. Therefore, gateways can affect network performance and quality of service (QoS), which is why their placement constitutes a significant problem. On the one hand, gateways need to be deployed all around the Earth, because satellite networks generally have global coverage and provide services worldwide. However, the number of gateways needed in a certain satellite network is undetermined. More gateways may provide better access services but can also predictably generate higher costs. On the other hand, gateways should be located at appropriate sites to optimize system performance for a given fixed satellite constellation [9]. More gateways can be reasonably deployed in places with higher-demand for Internet services while fewer can be justified in places like oceans and deserts. It also makes sense to deploy gateways close to key Internet exchange points where a huge proportion of Internet traffic is exchanged.

Hence, we propose here the gateway placement optimization (GPO) problem for LEO satellite networks, which is a new method for obtaining the optimal gateway placement scheme while minimizing investment, reducing implementation costs, and maintaining the preset QoS. We first formulate a GPO model with multiple objectives and constraints. One objective

Manuscript received July 31, 2020. ©2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0502402, and by the Agencia Estatal de Investigación of Spain under project PID2019-108713RB-C51/ AEI/10.13039/501100011033.

Jianming Guo, Lei Yang, Xiaoqian Chen, and Xianqi Chen are with the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: gjm08110@hotmail.com, craftyang@163.com, chenxiaoqian@nudt.edu.cn, chenxianqi12@nudt.edu.cn).

David Rincón and Sebastià Sallent are with the Department of Network Engineering, Universitat Politècnica de Catalunya (UPC), Castelldefels, 08860, Barcelona, Spain (e-mail: drincon@entel.upc.edu, sallent@entel.upc.edu).

concerns the network performance, which is evaluated by the load balancing index, since it can be estimated effectively and is not susceptible to other factors such as routing algorithms among satellites. The traffic load of a network's nodes can be estimated by several different methods [10], of which a gravity model is an effective and simple approximation method; thus, it is used to model commodity exchanges and has been applied in network applications [11]. We adopt this model to obtain the average traffic load of gateways and satellites. Moreover, a satellite has a limited satellite-ground bandwidth and a fixed number of antennas. Thus, the satellite's number of simultaneous gateway connections has a maximum limit, which we regard here as constraints. We also take into account the interference between gateway-to-satellite links when gateways are close to each other. Finally, as GPO constraints are nonlinear, it is difficult to use integer linear programming methods. So, we instead adopt the discrete version of the particle swarm optimization (PSO) algorithm, which is widely used and has been verified as an effective way for dealing with complex problems that have nonlinear objectives and constraints.

The rest of the paper is organized as follows. Section II briefly describes the state of the art in terms of three aspects: network GPO, traffic estimation, and the PSO algorithm. Section III illustrates the GPO problem by modeling it as a nonlinear, multi-objective, and multi-constraint combination optimization problem. Section IV describes the gravity model we use to compute the traffic load of gateways and proposes a method for calculating the constraints in the GPO model. In Section V, we modify the original discrete PSO algorithm in order to accelerate the GPO problem-solving process, and we discuss the algorithm analysis of discrete PSO parameters. Section VI presents case study on real satellite networks and gives the optimization results. Finally, Section VII concludes this paper.

II. Related Works

A. Gateway Placement

The gateway placement problem originally emerged in terrestrial networks such as wireless mesh and sensor networks. Existing works mainly adopt the integer linear programming model to find the optimal solution, with their objectives usually being to minimize the number of gateways [12], maximize network capacity [13], or minimize energy consumption [14]. In terms of constraints, QoS performance is sometimes considered in regard to parameters such as packet loss rate and average delay [12], [15], [16]. In these studies, researchers propose polynomial time algorithms to reduce computational complexity when the network scale increases. However, the problem is more complex when satellite mobility and handover of communication links are involved. Hence, the methods proposed above will not be effective in LEO satellite networks.

Previous studies on satellite networks seldom concern the GPO problem. To the best of our knowledge, it was first discussed in 2018 by Cao et al. [17], [18], who implemented a scenario with one GEO satellite and several gateways. Their aim was to find an optimal set of gateways from terrestrial nodes that maximize the weighted average network reliability or minimize network latency under capacity constraints. When Liu et al. [19] proposed the gateway placement problem in SDN-enabled 5G-satellite integrated networks, their objective was the network latency, for which they used an approximate method based on simulated annealing to solve the problem. Yang et al. [20] proposed a k-means method to solve the joint placement problem of controllers and gateways. In a word, the optimization methods above are worth learning, but their gateway placement models seem to be a little simple.

As far as we know, most existing works consider only GEO satellites and neglect satellite mobility. For this reason, their methods cannot be adopted in LEO constellations. Therefore, the delay between satellite and gateways is constant, and the GPO problem becomes irrelevant to satellites. Furthermore, the optimization objectives and constraints in existing works rarely involve satellite communications characteristics like bandwidth limitations; thus introducing more realistic constraints could improve the gateway placement scheme.

B. Traffic Estimation

Previous researchers have investigated methods for estimating node-to-node Internet traffic. Roughan et al. [21] first proposed the gravity model to derive an approximate traffic matrix of the Internet backbone. Medina et al. [22] compared three traffic estimation methods, namely linear programming, Bayesian inference approach, and expectation maximization algorithms. They further proposed a new method by introducing the choice model, whose form is similar to the gravity model. Gunnar et al. [23] also compared different methods of traffic estimation, including the gravity model, which they emphasized should be combined with statistical approaches to obtain better results.

Zhang et al. [11] combined the network tomography method with the gravity model and proposed a new method named tomogravity, which requires measuring link loads in order to more accurately infer traffic matrices in IP networks. The method consists of two steps: first, edge link load data is used to obtain a reference solution; second, quadratic programming finds the solution in the space of those admitted by the tomography model closest to the gravity model's reference solution. Rahman et al. [10] compared the three methods of tomogravity, entropy maximization and linear programming, and they found that the tomogravity method performs best for estimating the traffic matrix.

Although the tomogravity method provides the best traffic matrix estimates, the unavailability of real traffic measurements in satellite networks forces researchers to adopt the basic gravity model. Chen et al. [24] used the gravity model to estimate the inter-satellite traffic according to user density and host density. Yang et al. [25] and Wu et al. [26] divided the Earth's surface into 288 areas, then used world population density to represent the traffic demand within each area. They adopted the gravity model to estimate the traffic between two areas, then calculated the traffic demand of satellites by attaching each area to the closest satellite. We will follow the same approach and use the basic gravity model to estimate the traffic matrix in satellite networks. While using the more accurate data derived from the tomogravity method could obtain better results in future developments,this would incur no change in any of the other steps in our proposal.

C. Discrete PSO Algorithm

The PSO algorithm is a stochastic population-based optimization method first proposed by Kennedy et al. [27]. With a strong global searching ability, PSO can deal with complex nonlinear optimization problems. Since the original PSO is designed for continuous space, discrete versions of PSO have been developed to satisfy the special requirements of discrete issues. One problem in discrete PSO concerns the update process of the particle's velocity and position. To solve this problem, a discrete PSO method was also proposed by Kennedy et al. [28], who used a sigmoid function strategy to control the update process of the particle's position in binary PSO problems. Similarly, other strategies were also proposed, like multi-phase [29] and angle modulation [30] .

Jarboui et al. [31] proposed a combination PSO algorithm to solve a certain project scheduling problem. The core of the algorithm is to define the distance between the current solution and the particle's best solution up to that point, as well as the global best solution. Based on this definition, the particle's position and velocity are updated. Chen et al. [32] proposed a set-based PSO algorithm, where they used a potential set and the probabilities used to represent the particle's velocity. A particle can learn the possibility from each element of the set and thus moves to a new position. Furthermore, Chen et al. [33] provided a survey of the discrete PSO, especially the set-based PSO algorithm. Shen et al. [34] used the bi-velocity to represent the possibilities of being 0 and 1 in the binary domain, then applied the algorithm to multicast routing problems in communication networks. To sum up, all the discrete PSO methods seem to be potentially effective for the GPO problem, although the bi-velocity discrete PSO might match the GPO problem best due to the velocity definition that concerns possibilities.

III. Gateway Placement Optimization Problem

The architecture of a typical LEO satellite network is depicted in Fig. 1, which includes a LEO constellation of hundreds of satellites, several gateways, and telemetry, tracking and control (TT&C) stations. LEO satellites, which usually orbit at altitudes of hundreds to over a thousand kilometers, are regarded as switches, and they forward data packets from/to users and Internet servers. Users consist of mobile devices using specific modems, vehicles, aircraft, sensors and actuators, among others. The TT&C stations that are used for satellite management and maintenance will not be discussed further, since their locations are not necessarily optimized in terms of traffic, which is another problem.

There are three kinds of communication links, namely inter-satellite links (ISLs), user-satellite links (USLs) and gateway-satellite links (GSLs). ISLs are established between two adjacent satellites in either the same orbit or different orbits. ISLs are often radio (and sometimes optical) links, and they do not necessarily exist in previous satellite networks. In our model, however, we will take ISLs into consideration and simply assume the use of high-rate laser ISLs with enough transmission capacity, because we do not focus on inter-satellite data forwarding. GSLs and USLs are nowadays often designed in Ka and Ku band [35] in order to obtain a high data transmission rate. Users and gateways can set up connections with satellites within a line of sight through, respectively, USLs and GSLs. These satellites are called source or destination nodes, as shown in Fig. 1. Two users can communicate with each other through one or several satellites as well as USLs and ISLs, whereas communications between a user and servers that do not connect to the satellite network will involve gateways (i.e., the Internet backbone). With the direct radiating array (DRA) forming multiple beams [1], a satellite can provide Internet services through gateways to multiple users within its coverage area. Meanwhile, each satellite with several steerable gateway antennas can connect to multiple gateways. It should be noted that the coverage area of a satellite will change with time as it orbits around the Earth, and thus gateways within the coverage will change too.

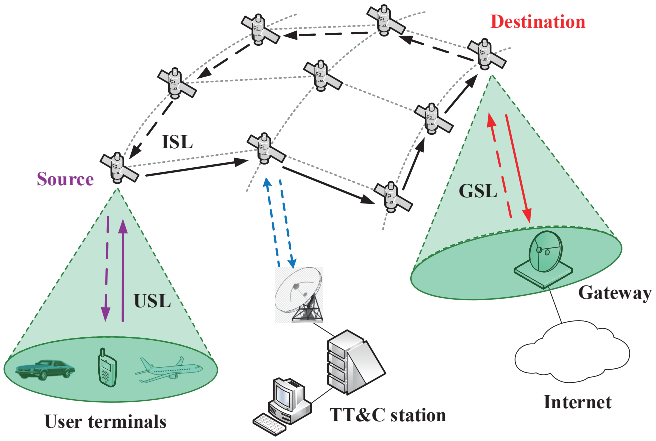


Fig. 1. Architecture of a typical satellite network.

A. Design Variables

Our aim is to establish the best location and the minimum number of gateways that will minimize investment and implementation costs while maintaining the preset performance. Therefore, the design variables are the gateway placement scheme, and they are determined by an alternative set of gateways. Assume a set of all possible gateway locations in which any two elements are different, meaning that two gateways cannot be in the same place. Thus, given a certain gateway placement scheme, any gateway in the potential set has only two states, (i.e., existence or non-existence), which suggests that the design variables have a binary form.

Supposing that the total number of possible gateways is N, we define an N-dimensional vector of 0-1 elements $\boldsymbol{x} = (x_1, x_2, ..., x_N)$ as the design variables. If gateway j $(1 \leq j \leq N)$ exists, then $x_j = 1$; otherwise, $x_j = 0$. If m gateways are

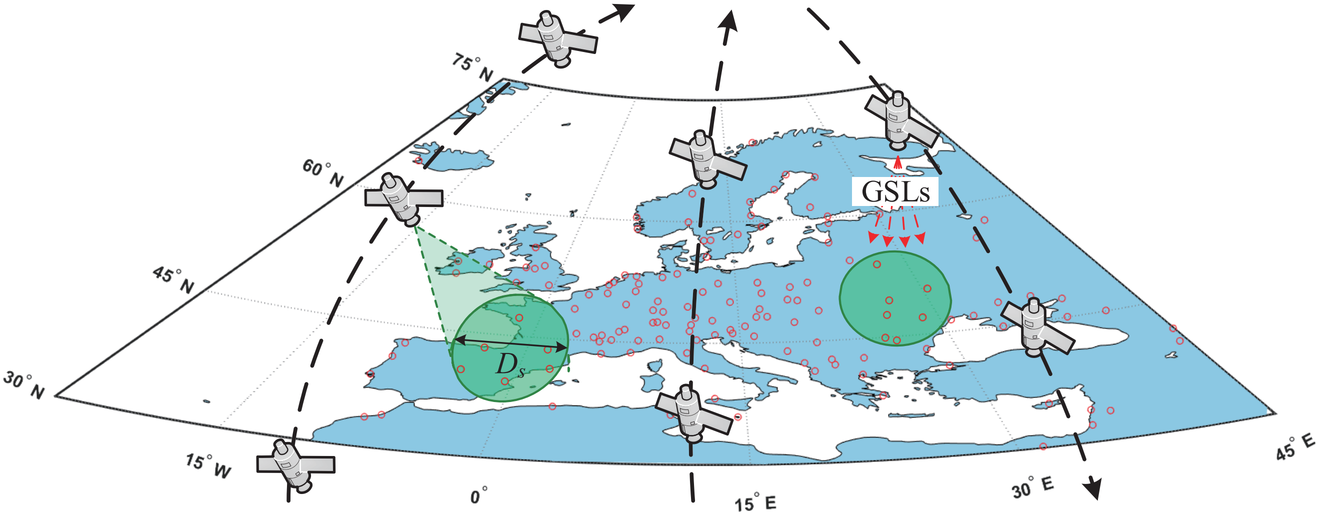


Fig. 2. Satellite's coverage area over gateways in Europe. Gateways are depicted as red circles, and each satellite has four GSLs represented by red dashed lines.

chosen, we have

$$\sum_{j=1}^{N} x_j = m \tag{1}$$

where $m \in \mathbb{N}$, $0 \leq m \leq N$. Therefore, an integer set $\boldsymbol{I}$ can be introduced to indicate the existence of gateways, i.e., $\boldsymbol{I} = \{i_k \in N, 1 \leq k \leq m \mid x_{i_k} = 1\}$.

B. Optimization Objectives

The optimization objectives can be derived from two aspects, namely the network performance and implementation costs. The latter can be evaluated easily by the total number of gateways; in other words, fewer gateways usually incur less investment and implementation expenditure. Thus, this objective can be the sum of the design variables as described in (1). In contrast, network performance is more complex to estimate, since it can be evaluated by multiple QoS parameters, such as latency, throughput and load balancing. Due to the highly dynamic topology of satellite networks, it is difficult to obtain the real end-to-end delay and network throughput without specifying the routing algorithm. furthermore, the load balancing is a desirable property that can be estimated effectively. Thus, we adopt it as an evaluation criterion for the suitability of our approach. That is to say, the traffic load of different satellites needs to be as balanced as possible to avoid-or at least minimize-network congestion.

We denote T_j as the average amount of traffic transported through each gateway, where $1 \leq j \leq N$, and T_j is a function of $\boldsymbol{x}$. It is only if gateway j exists that T_j will have a non-zero value. We set D_T as the absolute deviation of gateway traffic, which can be derived as

$$D_T = \frac{1}{m} \sum_{j \in \boldsymbol{I}} \left| T_j - \bar{T} \right| \tag{2}$$

where $\bar{T} = \frac{1}{m} \sum_{j=1}^{N} T_j$ is the average traffic of all the gateways. Therefore, the objective functions have the following normalized form,

$$f_1(\boldsymbol{x}) = \frac{D_T}{\bar{T}} \tag{3}$$

$$f_2(\boldsymbol{x}) = \frac{1}{N} \sum_{j=1}^{N} x_j \tag{4}$$

C. Constraints

On the one hand, it is better to locate gateways at sites where they can provide higher traffic bandwidth to users. On the other hand, each satellite has a limited bandwidth and a limited number of antennas in terms of GSLs, which imposes restrictions on the maximum number and the total traffic load of gateways that can simultaneously link to a satellite. In addition, the GSL interference should be avoided by ensuring that not too many gateways gather in a small region. We define the minimum distance between two gateways as $D_{\min}$ and denote the distance between any two gateways i and j as δ_{ij}. Since δ_{ij} should be as large as possible, the constraint can be formulated as

$$g_1(\boldsymbol{x}) = D_{\min} - \delta_{ij} \leq 0, (\forall i, j \in \boldsymbol{I}, i \neq j) \tag{5}$$

Each satellite has a restricted coverage area with a diameter of D_s. We assume that the number of gateway antennas on a satellite is C_s and that the limited GSL bandwidth is B_s. An example of a satellite's coverage area is depicted in Fig. 2, where a red circle represents a gateway, and a satellite can cover several gateways in Europe. Thus, the total traffic load of all the gateways within a satellite's coverage area should be less than B_s, and the total number of them should be fewer than C_s under the hypothesis that one antenna can connect to only one gateway.

Gateways within a certain satellite coverage area are a non-zero subset of $\boldsymbol{I}$, and it can be denoted as $\boldsymbol{U} = \{u_1, u_2, ..., u_k\}$, where $k \leq m$. As a result, the constraints can be formulated as follows

$$g_2(\boldsymbol{x}) = \sum_{j \in \boldsymbol{U}} T_j - B_s \leq 0, (\forall \boldsymbol{U} \subset \boldsymbol{I}, \boldsymbol{U} \neq \emptyset) \tag{6}$$

$$g_3(\boldsymbol{x}) = \sum_{j \in \boldsymbol{U}} x_j - C_s \leq 0, (\forall \boldsymbol{U} \subset \boldsymbol{I}, \boldsymbol{U} \neq \emptyset) \tag{7}$$

It should be noted that $\boldsymbol{U}$ in the two constraints may not be the same set, as multiple gateways, as a whole, may have a lower traffic load whilst a single gateway in some regions can possibly surpass the satellite's GSL bandwidth.

D. Formulation of the GPO Problem

As demonstrated above, the GPO problem can be summarized as

$$\begin{aligned} &\text{find } \boldsymbol{x} = (x_1, x_2, ..., x_N) \\ &\min \begin{cases} f_1(\boldsymbol{x}) = \frac{D_T}{T} \\ f_2(\boldsymbol{x}) = \frac{1}{N} \sum\limits_{j=1}^{N} x_j \end{cases} \\ &\text{s.t.} \begin{cases} g_1(\boldsymbol{x}) = D_{\min} - \delta_{ij} \leq 0, (\forall i, j \in \boldsymbol{I}, i \neq j) \\ g_2(\boldsymbol{x}) = \sum\limits_{j \in \boldsymbol{U}} T_j - B_s \leq 0, (\forall \boldsymbol{U} \subset \boldsymbol{I}, \boldsymbol{U} \neq \emptyset) \\ g_3(\boldsymbol{x}) = \sum\limits_{j \in \boldsymbol{U}} x_j - C_s \leq 0, (\forall \boldsymbol{U} \subset \boldsymbol{I}, \boldsymbol{U} \neq \emptyset) \end{cases} \end{aligned} \tag{8}$$

The formulation contains two objectives: $f_2(\boldsymbol{x})$ takes an integer and can be fixed to a certain value for the sake of simplification, if the GPO problem is too difficult to solve; thus, only $f_1(\boldsymbol{x})$ will exist. We elaborate on this in detail in Section V-C, but for consistency we will keep $f_2(\boldsymbol{x})$ here. Seemingly, we need to traverse all the values of i, j and $\boldsymbol{U}$ to meet these constraints, which is indeed inefficient for calculation. Hence, we transform them into simpler formulations.

To modify $g_1(\boldsymbol{x})$, we define k_d as the number of gateway pairs that are closer than $D_{\min}$, the minimum distance. If $k_d = 0$, the constraint function $g_1(\boldsymbol{x})$ will be satisfied. In order to make the constraint gradient descent, we calculate the average distance of the gateways pairs as $\bar{d} = \frac{1}{2k_d} \sum_{\delta_{ij} \leq D_{\min}} \delta_{ij}$, where δ_{ij} is the distance between gateway i and j, and the summation is divided by 2 because δ_{ij} is counted twice. Thus, $g_1(\boldsymbol{x})$ should decrease to 0, and the first constraint becomes

$$g_1(\boldsymbol{x}) = k_d(D_{\min} - \bar{d}) = 0 \tag{9}$$

To simplify $g_2(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$, we calculate the maximum of the total traffic load and that of the number of gateways within a satellite coverage area, and the constraints can be rewritten as

$$g_2(\boldsymbol{x}) = T_{\max} - B_s \leq 0 \tag{10}$$

$$g_3(\boldsymbol{x}) = U_{\max} - C_s \leq 0 \tag{11}$$

where $T_{\max} = \max\{\sum_{j \in \boldsymbol{U}} T_j\}$ and $U_{\max} = \max\{\sum_{j \in \boldsymbol{U}} x_j\}$ for $\forall \boldsymbol{U} \subset \boldsymbol{I}$.

Although various multi-objective optimization methods exist, it is still difficult to propose an effective and suitable algorithm for a special case. Usually, this nonlinear combination optimization problem with multiple objectives and complex constraints can be transformed into a single-objective non-constraint model by comprehensively utilizing the weighted sum method and penalty function method.

Since $g_1(\boldsymbol{x})$ equals 0 when the first constraint is met, it will have no effects on the overall objective. $g_2(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$ will be negative if these two constraints are met. In some cases, $g_2(\boldsymbol{x})$ should be as small as possible in order to reduce the utilization rate of satellite bandwidth, even though $g_2(\boldsymbol{x}) \leq 0$ is satisfied. Hence, $g_2(\boldsymbol{x})$ stays in the same form and will be regarded as an objective. By contrast, $g_3(\boldsymbol{x})$ is more like a real constraint, and it is enough to satisfy $g_3(\boldsymbol{x}) \leq 0$ instead of reducing the utility of antennas. Here, we adopt a dimensionless penalty function $\varphi(x)$ described in [36] for $g_3(\boldsymbol{x})$, which is defined as

$$\varphi(x) = \begin{cases} a(e^{bx} - 1), x > 0 \\ 0, x \leq 0 \end{cases} \quad (a, b > 0) \tag{12}$$

where the coefficients a and b are used to accelerate the decreasing process, and they will be given in the specific case. Therefore, $g_3(\boldsymbol{x})$ takes the following form:

$$g_3(\boldsymbol{x}) = \varphi(U_{\max} - C_s) \tag{13}$$

Thereby, the optimization model is now elaborated as

$$\begin{aligned} &\text{find } \boldsymbol{x} = (x_1, x_2, ..., x_N) \\ &\min F(\boldsymbol{x}) = w_1^o f_1(\boldsymbol{x}) + w_2^o f_2(\boldsymbol{x}) + \sum_{i=1}^{3} w_{i+2}^o \lambda_i g_i(\boldsymbol{x}) \end{aligned} \tag{14}$$

where w_i^o is either the weight coefficient or penalty factor, and λ_i is the normalization factor, namely $\lambda_1 = 1/D_{\min}$, $\lambda_2 = 1/B_s$, $\lambda_3 = 1$.

IV. Methods for Computing the GPO Model

In order to solve the GPO model, we have to compute the objective function $F(\boldsymbol{x})$. According to equations from (8) to (14), $F(\boldsymbol{x})$ involves several quantities including T_j, $D_{\min}$, $T_{\max}$, and $U_{\max}$. In this section, we propose methods to compute these quantities. Nevertheless, several steps still must be taken before calculation. First, in order to guarantee the computational precision, we need to discretize the Earth's surface and determine a minimum resolution of positions. Second, a potential location set of gateways should be determined. As stated before, the gravity model is the most logical choice for estimating the traffic matrix, so we adopt it to our case to calculate the traffic matrix between users and gateways. This allows us to obtain the traffic load of gateways, namely T_j, and then the traffic of a certain satellite by adding all the gateway traffic within its coverage area. Finally, in order to compute $D_{\min}$, $T_{\max}$, and $U_{\max}$, we need to find the relationship between a satellite coverage area and gateway positions, which is quite complex due to satellite movements.

A. Zone Unit and Resolution on the Earth's Surface

By discretizing the Earth's surface and determining a minimum resolution of positions, we can define the zone unit in order to avoid placement conflicts, which means that those gateways in the same zone unit can be reduced to one.

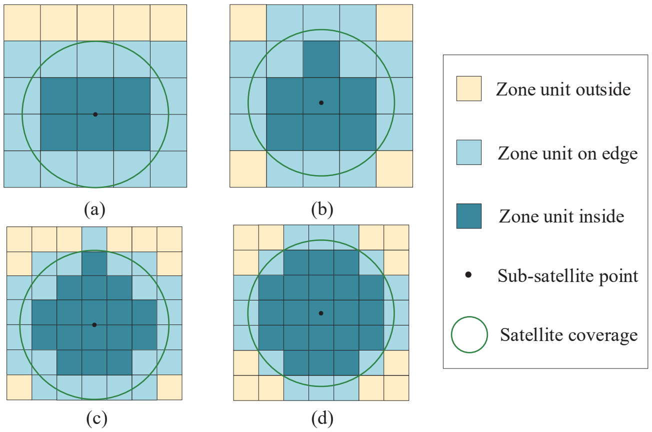


Fig. 3. Satellite coverage of different zone unit sizes. The sub-satellite point can be anywhere within a certain zone unit.

Moreover, the coverage area of each satellite will also have a clear boundary. Finally, because users are distributed around the world, dividing them into discrete units will be helpful for the computational process.

The Earth's surface can be discretized by different resolutions, thus resulting in different zone unit sizes. A high resolution usually generates high computational complexity, whilst a low resolution (or, in other words, a big zone unit) makes the coverage area of a satellite ambiguous. Therefore, it is reasonable to determine a relatively low but appropriate resolution. A zone unit is regarded as being inside the coverage area only if it is completely inside. In this way, a zone unit lying at the edge of a satellite coverage area is not deemed to be inside. As shown in Fig. 3, for the same satellite coverage area, the bigger zone units in (a) and (b) can result in fewer inside zone units compared with the smaller zone units in (c) and (d). In order to determine the division resolution and then the size of the zone unit, we need to evaluate the covering effectiveness of a satellite coverage area. Assuming that the coverage area of a certain satellite is C_a, we define the coverage ratio as

$$\zeta = \frac{z_{in}}{z_{tot}} \tag{15}$$

Here, z_{in} is denoted as the number of zone units inside C_a, and z_{tot} is the total number of zone units that are inside and on the edge of C_a. Since ζ might change when a satellite is moving (as shown in Fig. 3), the satellite in (a) and (b) can cover different numbers of zone units, similarly as in (c) and (d). Thus, we use the average value of ζ for each C_a to evaluate its coverage ratio. As the zone units are identical, we locate the sub-satellite point at different positions within a certain zone unit and calculate each ζ to obtain the average value.

According to the geometric relation of the satellite coverage area, C_a is determined by the orbit altitude h and the minimum ground elevation angle E. Thus, the diameter of the coverage area, namely D_s, is

$$D_s = 2 \arccos[R_e/(h + R_e) \cos E] - 2E \tag{16}$$

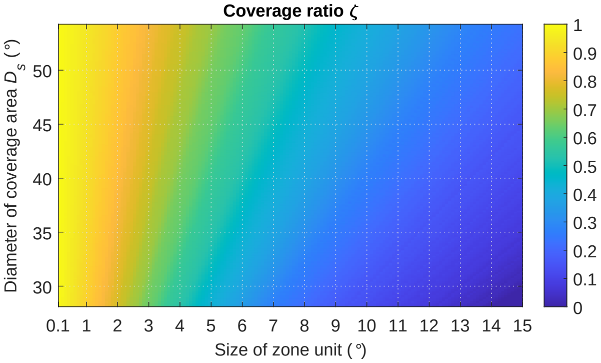


Fig. 4. Coverage ratio ζ varies with the sizes of the satellite coverage area and the size of zone units. The color represents the value of ζ.

where D_s is measured in the geocentric angle, and R_e is the Earth's equivalent radius. In general, the orbit altitude of LEO satellites can vary from 500 km to 1500 km, and E is often a small angle.

The length of a zone unit is set equal in latitude and longitude. We calculate the coverage ratio ζ under different coverage areas as well as different zone unit sizes ranging from 0.1° to 15°. Results are shown in Fig. 4, where E is set to 10°, and thus D_s will vary from 28° to 54°. It can be seen that ζ can be more than 90% only when the size of the zone unit is smaller than 1°, regardless of D_s, as it is shown on the left x-axis. The color represents the value of ζ. Therefore, the size of a zone unit is set to 1° in this paper.

B. Potential Location Set of Gateways

There is no need to consider all the positions on Earth's surface. For instance, ocean regions are not very suitable for gateway placement. Furthermore, as gateways are regarded as edge routers between satellite networks and the Internet, it is reasonable to locate them near sites of Internet exchange points (IXPs) for lower latency, fewer bandwidth resources and less cost. An IXP is a kind of Internet infrastructure that offers a shared switching fabric where telecom and service providers can exchange traffic with one another, once they have established peering connections between them [37]. Thus, a set of IXPs can be regarded as the potential location set of gateways.

The worldwide dataset of IXPs obtained from Packet Clearing House (PCH) [38], which is a global directory of IXPs, is used as a potential location set of gateways. According to PCH, there are originally 1022 IXPs in total around the world. However, only 516 of them are different and we therefore use these after eliminating the repetitive IXPs located at the same sites. Moreover, given the aforementioned aspects of zone unit resolution, multiple IXPs in the same zone unit should be reduced to one. Nonetheless, the repeatability of gateways at the same site is used as the weight factor $\boldsymbol{w} = (w_1, w_2, ..., w_N)$, a $1 \times N$ vector. For instance, if the dataset indicates that 5 IXPs are in Beijing, the weight of the IXP located in the Beijing's zone unit is 5. Ultimately, 471 IXPs are retained, as shown in Fig. 5.

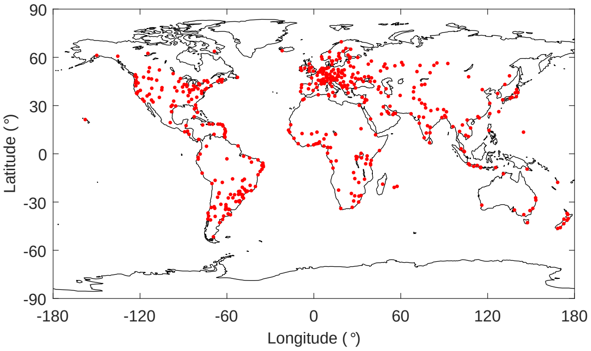


Fig. 5. The distribution of potential gateway sites, 471 in total. Gateways are indicated by red dots.

C. Traffic Estimation Methods

A traffic matrix is the point-to-point traffic of each pair of nodes in a network. It can be estimated by the linear relationship between each link and each pair of nodes. A network with n nodes and l links has $p = n(n-1)$ pairs of distinct nodes that may communicate with each other. We use t_{ij} to denote the traffic from node i to node j directly or through other nodes, and thus $\boldsymbol{T} = [t_{ij}]_{n\times(n-1)}$ is called the traffic matrix, which is usually represented in a vector form, namely $\boldsymbol{t} = (t_1, ..., t_p)^{\mathrm{T}}$. The traffic on l links can be measured and denoted as $\boldsymbol{y} = (y_1, ..., y_l)^{\mathrm{T}}$, so the relation between t and y is $\boldsymbol{y} = \boldsymbol{\Omega t}$, where $\boldsymbol{\Omega} = [\Omega_{ij}]_{l\times p}$ represents the fraction of traffic volume for node pair j crossing link i. The aim of network traffic matrix estimation is to determine the traffic vector $\boldsymbol{t}$, given the traffic link data $\boldsymbol{y}$ [10].

Several techniques exist for estimating the traffic matrix, of which the previously mentioned gravity model is an effective method in a topology-determined network. Derived from Newton's law of gravitation, the gravity model is used to predict the movement of people, information, and commodities between cities and continents [21]. Accordingly, the traffic from node i to node j, namely t_{ij}, can be estimated by the following equation,

$$t_{ij} = \frac{R_i A_j}{f_{ij}} \tag{17}$$

Here, R_i represents the repulsive factors associated with leaving from i while A_j is the attractive factors associated with coming to j, and f_{ij} is a friction factor associated with nodes i and j. In terms of network traffic estimation, these quantities are determined by the traffic data of nodes i and j. The traffic matrix t_{ij} can be described as a proportion of the total traffic amount coming from node i, and the proportion factor is related to the incoming traffic amount of node j and the geographic distance between them. Thus, we have the following equation:

$$t_{ij} = \frac{1}{\sum_{k\neq i} t_k^D / d_{ik}^{\alpha}} \cdot \frac{t_i^S t_j^D}{d_{ij}^{\alpha}} \tag{18}$$

where, t_i^S is the total amount of traffic originating from node i, representing its repulsive factor, and t_j^D is the total amount of traffic sent to node j, representing the attractive factor. d_{ij} is the spherical distance between nodes i and j. α is the index factor affecting the attenuation rate of the attractive factor t_j^D along with the distance d_{ij}.

In our model, all the edge nodes in a satellite network are divided into two independent parts, namely the users and the gateways. This means that node i and j in the gravity model are in two different sets. We use (18) to estimate the traffic between users and gateways.

D. Determining the Repulsive and Attractive Factors

The repulsive factor t_i^S, namely the traffic demand of users in zone unit i, is proportional to the population density there. Thus, it can easily be obtained by establishing a traffic demand equation. However, the attractive factor t_j^D, is more complicated to calculate. On the one hand, it is difficult to determine the effective region of gateways. Unlike a certain zone unit with a specific area, which is associated with its repulsive factor, gateways on the ground do not have clear service boundaries. We use a Voronoi diagram to divide the effective regions of gateways. On the other hand, the attractive factor is not related only to the population within the gateway's effective region. As Internet traffic is more likely drawn to areas with more data centers or servers, it is reasonable to use the region's number of IXPs for correcting the gateway's attractive factor. In this way, some regions with high population densities like Africa and Southeast Asia might have a relatively small attractive factor due to having fewer IXPs compared with Europe and North America.

As a result, we need to obtain the population density, the Voronoi division, the population, and the number of IXPs within the gateway's effective region. Then, the repulsive/attractive factor can be described as follows:

$$\begin{cases} t_i^S = \epsilon\gamma p_i^S \\ t_j^D = w_j p_j q_j \end{cases} \tag{19}$$

where ϵ is the external traffic demand of a single user, γ is a scale factor, and p_i^S is the total population pertaining to zone unit i. w_j is a component of the gateway's weight factor $\boldsymbol{w}$. p_j and q_j are, respectively, the total population quantity and the number of IXPs within the effective region of gateway j.

ϵ often represents the average value of traffic, ranging from 2.4 Kbps to 2 Mbps for satellite networks. γ is mainly determined by three aspects, namely user ratio, traffic profile and multiplexing gain. Generally, a satellite network has a certain market share compared with terrestrial communication operators, and the percentage of the total population using satellite services in a certain region is predicted to be around 10%. The traffic of each user will be forwarded to either a gateway (thus, the Internet backbone) or to another user that is directly connected to the satellite network. Therefore, the amount of the former kind of traffic takes up a fraction of the total traffic, which is assumed to be from 1/5 to 1/10. Since the traffic volume is not always entirely occupied, the ratio between peak period and average period is about 2:1. Furthermore, for the average period, only 1/5 of users take up the channel capacity. Considering the multiplexing mechanism in satellite networks, users can share channels without losing the communication quality, and it is reasonable to set the

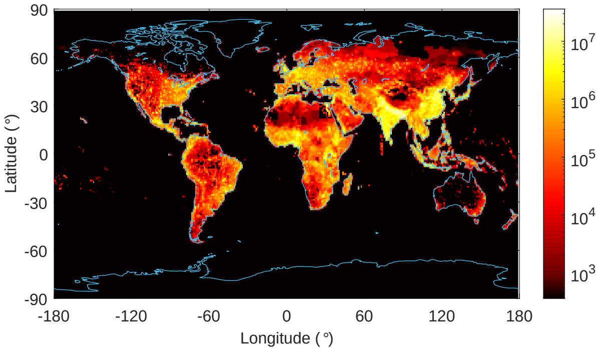


Fig. 6. The world population density distribution in 2020. The color represents the population in an area of 1° by 1°.

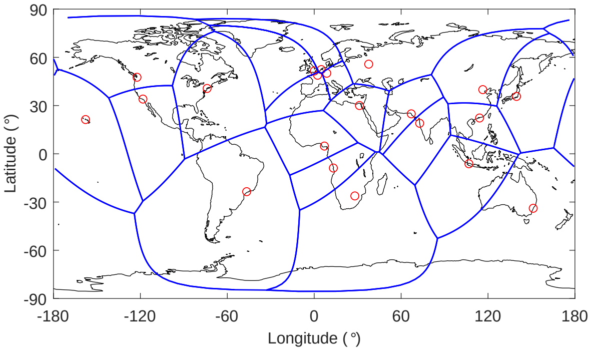


Fig. 8. Spherical Voronoi diagram of the Earth's surface.

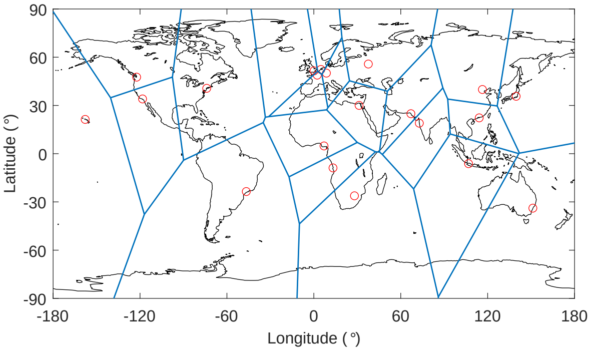


Fig. 7. Planar Voronoi diagram of the Earth's surface.

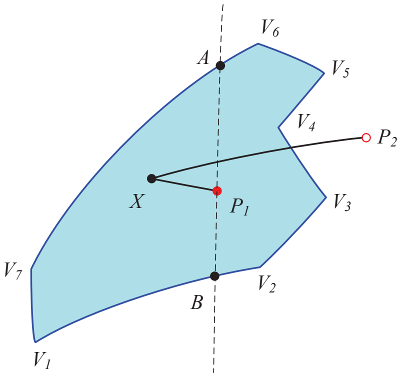


Fig. 9. Points inside or outside a spherical polygon.

multiplexing gain from 5:1 to 10:1 for Internet data services. Therefore, γ is assumed to be about 10^{-4} in this work. The following paragraphs in this section will focus on the calculation of p_i^S, p_j and q_j.

1) World Population Density: The world population density data with resolution of 1° are generated from [39], which is the first version in 2020. We use a log scale map to illustrate it as shown in Fig. 6. We denote $\boldsymbol{P}$ as the world population density matrix, and p_i^S will be an element of $\boldsymbol{P}$.

2) Voronoi Diagram: A Voronoi diagram is usually used to divide a plane into several disjointed regions associated with the objects of a given set. The objects, called seeds, are often a finite number of points on the plane. In a Voronoi diagram, points in a certain region are closer to the region's seed than to other seeds.

We calculate the planar Voronoi diagram of the Earth's surface, given a set of gateways, as shown in Fig. 7. Here, 21 IXPs with nearly the highest traffic are chosen as the reference set for our simulation. Results suggest that the planar Voronoi diagram cannot be adapted to the Earth's surface, because of the inconsecutive boundaries of the longitude at $-180°$ and $-180°$, as well as in polar regions.

We adopt spherical Voronoi diagram, thus converting the effective regions of gateways into spherical polygons. Each edge of a spherical polygon is a great circle arc, often the minor arc, determined by the two end vertices. Fig. 8 shows an example of a spherical Voronoi diagram calculated by the algorithm in [40], given the same set of gateways. As we can see, the boundaries of spherical polygons are much more reasonable.

3) Points within a Spherical Polygon Region: To calculate the attractive factor t_j^D of gataway j, we need the total population and the number of IXPs within its effective region, that is, a spherical polygon. This problem can be regarded as locating points on a spherical surface relative to a spherical polygon. [41] proposed a useful method for locating a single point on a sphere. However, since we need to locate a large number of points, especially for all the zone units inside the spherical polygon, it would be inefficient to judge all the points one-by-one. We modify the original method and propose the positional range in spherical-polygon (PRIS) algorithm, as illustrated in Algorithm 1, to determine massive points inside or outside a certain spherical polygon.

Let us assume a spherical polygon S with a set of vertices V_s, and the inner region of it is confined by its vertices that are sorted anti-clockwise from V_1 to V_7, as shown in Fig. 9. Here, P_j ($j = 1, 2, \dots$) is a point on the spherical surface, and a reference point X inside S is also needed to determine whether a point lies inside (including being on the edge) or outside S. According to [41], the key to this problem is to find how many times that XP_j will cross the boundary of S, where XP_j is the minor arc joining X and P_j. If P_j is inside S, XP_j will cross the boundary of S an even number of times; otherwise, an odd number of times.

Once we determine whether an arbitrary point is inside or outside a spherical polygon, we can deal with massive points by the PRIS algorithm. The core function of the algorithm is to calculate the latitude range of the spherical polygon for a given longitude. The PRIS algorithm first generates the longitude range of the spherical polygon S, and traverses the entire longitude to obtain its latitude range. For a certain longitude as shown in Fig. 9, the PRIS algorithm regards the north pole (or south pole, if calculating for the southern hemisphere) as the reference point X, and calculates the cross points A and B of S. If the north pole (or south pole) is outside S, there will be an even number of cross points. Otherwise, the longitude range will be from $-180°$ to $180°$, and only one cross point will exist for each line of longitude. Besides, if S crosses the $180°$ longitude, the longitude range will be different. As a result, the latitude range can be obtained from the coordinates of cross points.

Algorithm 1 Positional Range In Spherical-polygon.

```
Input: Spherical polygon vertices $V_s$, reference point $X$
Output: Latitude range, $R_{lat}$
if $S$ consists of the north pole or south pole then
    The longitude range of $S$ is $R_{lon} = [-180, 180]$
else
    Generate the longitude range of $S$, $R_{lon}$
end if
for all $i \in R_{lon}$ do
    Calculate the cross points between $i$ and $S$, $C_p(i)$
end for
Obtain the latitude range vector $R_{lat}$ by using $C_p$
```

E. Estimate the Traffic Load of Gateways

As indicated above, we can calculate the traffic load of gateways based on several hypotheses. First, the gateways are located at IXP sites. Second, the traffic demand of a zone unit on the Earth is related to the population density in that location. Third, the traffic between two nodes in a network can be estimated according to the gravity model. Finally, the Voronoi diagram can be used to divide the effective region relative to a certain gateway.

The data traffic t_{ij} between user i and gateway j is illustrated in Fig. 10. Here, three or more satellites are involved in this process, with one covering the user and another the gateway. The traffic load of gateway j is then the sum of t_{ij} by index i,

$$T_j = \sum_i t_{ij} \tag{20}$$

The computing procedure of T_j is depicted in Fig. 11. The input variable is an N-dimensional vector of 0-1 elements $\boldsymbol{x} = (x_1, x_2, ..., x_N)$, indicating the existence of each gateway. The constant parameters required for the calculation procedure are summarized as follows. $\boldsymbol{G}$ is a $2 \times N$ matrix of gateway positions, where the first and second rows represent, respectively, latitude and longitude. $\boldsymbol{w}$ is the weight factor of gateways. $\boldsymbol{P}$ is a 180×360 matrix of world population density distribution. ϵ is a scalar of the user's previously defined external traffic demand that has been defined before. As it is shown in Fig. 11, the computing process has the following steps.

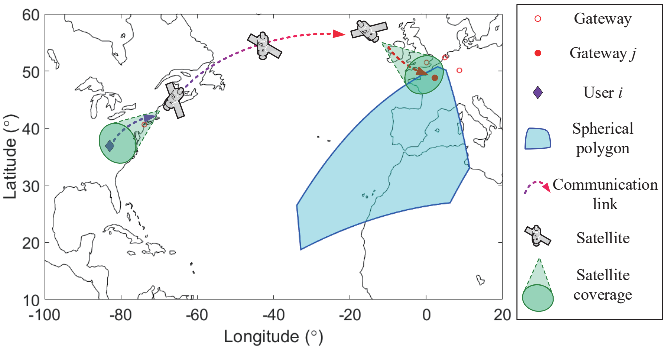


Fig. 10. Data forwarding process between user i and gateway j through satellites.

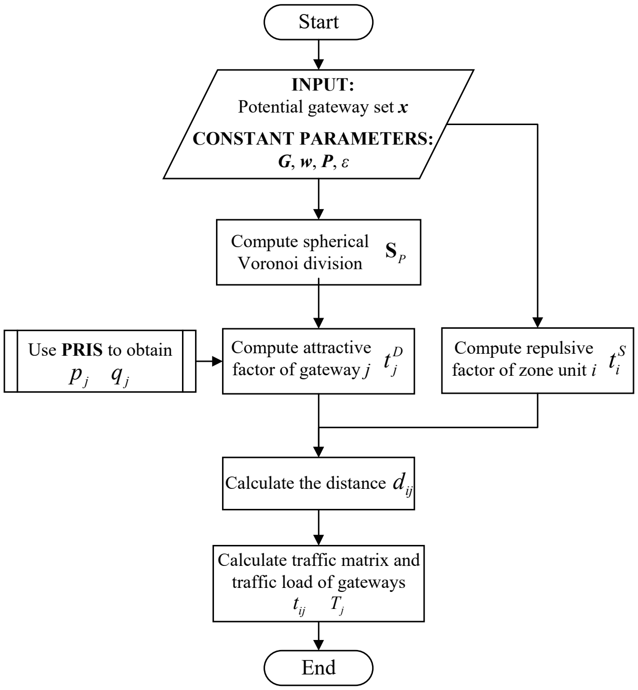

Start
INPUT:
Potential gateway set $\boldsymbol{x}$
CONSTANT PARAMETERS:
$\boldsymbol{G}$, $\boldsymbol{w}$, $\boldsymbol{P}$, ε
Compute spherical Voronoi division $\mathbf{S}_P$
Use PRIS to obtain p_j q_j
Compute attractive factor of gateway j t_j^D
Compute repulsive factor of zone unit i t_i^S
Calculate the distance d_{ij}
Calculate traffic matrix and traffic load of gateways t_{ij} T_j
End

Fig. 11. Computing procedure of the traffic load of gateways.

1) Calculate the total traffic t_i^S from zone unit i by $\epsilon\gamma\boldsymbol{P}$.

2) Use the spherical Voronoi diagram to obtain the Voronoi division S_P, a set of spherical polygons corresponding to each potential gateway.

3) Use the PRIS algorithm with input variables S_P to calculate all the zone units inside each spherical polygon of S_P, and then obtain the total population p_j as well as the total number of IXPs q_j within each spherical polygon associated with gateway j.

4) Calculate the attractive factor t_j^D of gateway j by (19).

5) Calculate the spherical distance d_{ij} between zone unit i and gateway j.

6) Calculate the user-to-gateway traffic t_{ij} by (18), and then the load of each gateway (i.e., T_j), can be calculated with (20).

We still take the reference set of 21 IXPs as an example to compute their traffic load. The result is shown in Fig. 12,

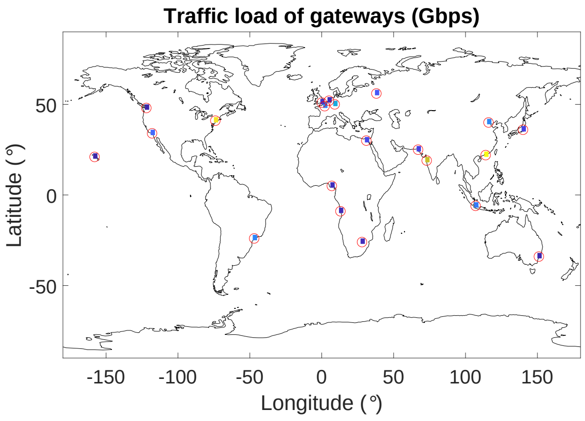


Fig. 12. Traffic load of the test gateways.

where $\epsilon = 100$ Kbps, and the index factor of distance in (18) is $\alpha = 1$.

F. Calculate the Constraints

As a satellite orbits around the Earth, its coverage area and gateways underneath it will change, which makes constraints $g_2(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$ complex to calculate. We use an example to illustrate the relation between satellite coverage area and gateways, as depicted in Fig. 13. Here, gateways can be simultaneously in different coverage areas of several satellites or that of the same satellite at different times. Gateways numbered 1 to 5 are within one satellite coverage area; gateways 4 to 8 and 6 to 10 are covered, respectively, by different satellites.

Evidently, we need to traverse first the position of sub-satellite points to calculate the total traffic within a coverage area, and then the constraints, which may cost huge computational resources. Nevertheless, we can simplify this problem using the number of gateways that a satellite can cover, which is denoted as k. Assume the total number of existing gateways is m, so a satellite can cover at most m gateways. We traverse k from 1 to m and calculate the traffic load of satellites, since this is unnecessary to calculate when $k = 0$. Therefore, the problem becomes how to find all the gateways within a satellite coverage area given a fixed k.

As described before, the integer set $\boldsymbol{I}$ indicates the existence of gateways, and a group of gateways within a certain satellite coverage area is denoted as $\boldsymbol{U}$, which is a none-zero subset of $\boldsymbol{I}$. Therefore, any elements of $\boldsymbol{U}$ must satisfy

$$\delta_{ij} \leq D_s, \forall i, j \in \boldsymbol{U} \tag{21}$$

where δ_{ij} is the distance between gateway i and j.

Thus, given an integer set $\boldsymbol{I}$, we define a symmetric matrix $\boldsymbol{R} = [r_{ij}]_{m \times m}$ as

$$r_{ij} = \begin{cases} 1, & i \neq j \text{ and } \delta_{ij} \leq D_s \\ 0, & i \neq j \text{ and } \delta_{ij} > D_s \\ 0, & i = j \end{cases} \tag{22}$$

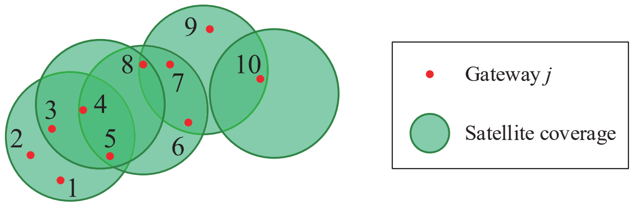


Fig. 13. The satellite can cover an area with different gateways.

Assume $\boldsymbol{U} = \{u_1, u_2, ..., u_k\}$, where $k \leq m$, so for any $\boldsymbol{U} \subset \boldsymbol{I}$ we will have a principal minor of $\boldsymbol{R}$ associated with $\boldsymbol{U}$, namely

$$\boldsymbol{R_U} = \begin{pmatrix} R_{u_1u_1} & R_{u_1u_2} & \cdots & R_{u_1u_k} \\ R_{u_2u_1} & R_{u_2u_2} & \cdots & R_{u_2u_k} \\ \vdots & \vdots & \ddots & \vdots \\ R_{u_ku_1} & R_{u_ku_2} & \cdots & R_{u_ku_k} \end{pmatrix} \tag{23}$$

At the same time, a reference matrix $\boldsymbol{R_{ef}}$ is defined as

$$\boldsymbol{R_{ef}} = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}_{k \times k} \tag{24}$$

Therefore, if $\boldsymbol{R_U} = \boldsymbol{R_{ef}}$, then gateways defined by $\boldsymbol{U}$ are within a satellite coverage area. For each k, we obtain all the $\boldsymbol{U}$ satisfying $\boldsymbol{R_U} = \boldsymbol{R_{ef}}$, and find the maximum value of $\sum_{j \in \boldsymbol{U}} T_j$. Then the constraint function $g_2(\boldsymbol{x})$ can be calculated. Similarly, $g_3(\boldsymbol{x})$ can also be calculated.

V. Modified Discrete PSO Algorithm

The GPO model described in (14) is a nonlinear combination optimization problem with multiple objectives that include several transformed constraints, for which integer linear programming algorithms are not appropriate. Therefore, we use heuristic methods including PSO algorithm to solve this optimization problem. Here, we adopt the discrete PSO algorithm to obtain the optimal gateway placement solution.

A. PSO Algorithm

At first, we elaborate on the mechanism of the general PSO algorithm according to [31]. In an N-dimensional PSO problem with a population size of K, particle i $(1 \leq i \leq K)$ has a position vector $\boldsymbol{x_i} = (x_{i1}, x_{i2}, ..., x_{iN})$ and a velocity vector $\boldsymbol{v_i} = (v_{i1}, v_{i2}, ..., v_{iN})$. During the optimization process, each particle updates its position and velocity in an iteration according to three aspects: the particle's inertia, its own experience (self-cognition), and the whole swarm's experience (social-influence). Therefore, the position and velocity are iterated as follows:

$$\begin{aligned} v_{ij} &\leftarrow \omega v_{ij} + c_1 r_{1j}(p_{ij} - x_{ij}) + c_2 r_{2j}(g_j - x_{ij}) \\ x_{ij} &\leftarrow x_{ij} + v_{ij} \end{aligned} \tag{25}$$

Here, x_{ij} is the jth dimension of $\boldsymbol{x_i}$, and we denote the maximum number of iterations as T. The historical best position of particle i is recorded as $\boldsymbol{p_i^{best}} = (p_{i1}, p_{i2}, ..., p_{iN})$, and the global best position so far among the population is also kept as $\boldsymbol{g^{best}} = (g_1, g_2, ..., g_N)$. ω represents the inertia weight and often linearly decreases with the number of iterations; therefore, we denote the value range of ω as $[\Omega_{start}, \Omega_{end}]$. c_1 and c_2 are acceleration coefficients associated with self-cognition and social-influence, respectively. r_{1j} and r_{2j} are random numbers uniformly distributed in $[0, 1]$.

If the independent variable $\boldsymbol{x_i}$ has a lower bound B_L and an upper bound B_U, there should be a procedure for verifying boundary conditions after the position and velocity are updated.

B. Discrete PSO Algorithm

As the independent variables in our model are binary integers for indicating the existence or non-existence of gateways, i.e., either 1 or 0, we introduce the bi-velocity discrete PSO algorithm proposed in [34]. The position vector of particle i $(1 \le i \le K)$, namely $\boldsymbol{x_i}$, has the same form as the general case:

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, ..., x_{iN}), \text{ where } x_{ij} = 0 \text{ or } 1 \tag{26}$$

Its velocity has a bi-velocity form:

$$\boldsymbol{v_i} = \begin{pmatrix} \boldsymbol{v_i^0} \\ \boldsymbol{v_i^1} \end{pmatrix} = \begin{pmatrix} v_{i1}^0 & v_{i2}^0 & \cdots & v_{iN}^0 \\ v_{i1}^1 & v_{i2}^1 & \cdots & v_{iN}^1 \end{pmatrix} \tag{27}$$

Here, v_{ij}^0,$v_{ij}^1 \in [0, 1]$. v_{ij}^0 is the possibility of $x_{ij} = 0$, whilst v_{ij}^1 is the possibility of $x_{ij} = 1$. The velocity updating rule is described as follows.

Assuming that $\boldsymbol{x_2}$ is a better position than $\boldsymbol{x_1}$ in terms of the fitness function, $\boldsymbol{x_1}$ should learn from $\boldsymbol{x_2}$. We need to calculate $\Delta\boldsymbol{v} = \boldsymbol{x_2} - \boldsymbol{x_1}$ as the updated increment. If $x_{2j} = x_{1j}$, then x_{1j} does not need to learn from x_{2j}, which means that $\Delta v_j^0 = 0$, $\Delta v_j^1 = 0$; or else x_{1j} needs to learn from x_{2j}, for which there are two possibilities: if $x_{2j} = 1$, then $\Delta v_j^0 = 0$ and $\Delta v_j^1 = 1$; if $x_{2j} = 0$, then $\Delta v_j^0 = 1$ and $\Delta v_j^1 = 0$. Furthermore, the linear operation rules of the velocity are

$$\boldsymbol{v_1} + \boldsymbol{v_2} = \begin{pmatrix} \max\{v_{11}^0, v_{21}^0\} & \cdots & \max\{v_{1j}^0, v_{2j}^0\} & \cdots \\ \max\{v_{11}^1, v_{21}^1\} & \cdots & \max\{v_{1j}^1, v_{2j}^1\} & \cdots \end{pmatrix} \tag{28}$$

$$\lambda\boldsymbol{v} = \begin{pmatrix} \lambda\boldsymbol{v^0} \\ \lambda\boldsymbol{v^1} \end{pmatrix} = \begin{pmatrix} \lambda v_{i1}^0 & \lambda v_{i2}^0 & \cdots & \lambda v_{iN}^0 \\ \lambda v_{i1}^1 & \lambda v_{i2}^1 & \cdots & \lambda v_{iN}^1 \end{pmatrix} \tag{29}$$

where $\max\{*\}$ means to take the maximum value for a certain element in a matrix.

When multiplying a coefficient by the velocity or adding two velocities, the result might exceed the value range. To meet the boundary conditions, we regulate that any element of $\Delta\boldsymbol{v}$ that is bigger than 1 should be set to 1. We define β as a critical coefficient for determining the change of positions. So, $\boldsymbol{v_i}$ and $\boldsymbol{x_i}$ are determined as follows:

$$v_{ij} = \max\{\omega v_{ij}, c_1 r_{1j} \Delta v_{ij}^p, c_2 r_{2j} \Delta v_{ij}^g\} \tag{30}$$

$$x_{ij} = \begin{cases} \text{rand}\{0, 1\}, & \text{if}(v_{ij}^0 > \beta \text{ and } v_{ij}^1 > \beta) \\ 0, & \text{if}(v_{ij}^0 > \beta \text{ and } v_{ij}^1 \le \beta) \\ 1, & \text{if}(v_{ij}^0 \le \beta \text{ and } v_{ij}^1 > \beta) \\ x_{ij}, & \text{if}(v_{ij}^0 \le \beta \text{ and } v_{ij}^1 \le \beta) \end{cases} \tag{31}$$

where $\Delta v_{ij}^p = p_{ij} - x_{ij}$ and $\Delta v_{ij}^g = g_j - x_{ij}$.

C. Modified Discrete PSO Algorithm

Since the independent variable $\boldsymbol{x}$ has a very high dimension, the solution domain is thus too large. Furthermore, the constraints are nonlinear and complex, which makes it difficult for the discrete PSO algorithm to explore the optimal solution. To this end, we propose using the modified discrete PSO (MD-PSO) algorithm. With this, we try to fix the number of gateways chosen randomly from the N-dimensional set, which means that $m = \sum_{j=1}^{N} x_j$ is constant, and then m should be traversed. The m gateways should be unique, i.e., $i \ne j$, $\forall i, j \in \boldsymbol{I}$. Therefore, the objective function $f_2(\boldsymbol{x})$ can be reduced and a new constraint should be imposed on the fixed number of chosen gateways, namely $G(\boldsymbol{x})$. Thus, the GPO model becomes

$$\begin{aligned} &\text{find } \boldsymbol{x} = (x_1, x_2, ..., x_N) \\ &\min\ F(\boldsymbol{x}) = w_1^o f_1(\boldsymbol{x}) + \sum_{i=1}^{3} w_{i+2}^o \lambda_i g_i(\boldsymbol{x}) \\ &\text{s.t. } G(\boldsymbol{x}) = \sum_{j=1}^{N} x_j - m = 0 \end{aligned} \tag{32}$$

The MD-PSO algorithm is illustrated in Algorithm 2, which is quite similar to the general PSO algorithm except for the verification process of the constraint $G(\boldsymbol{x})$. Among the inputs, N is the dimension of $\boldsymbol{x}$, m is the number of existing gateways, K is the population size, and T is the maximum number of iterations.

Algorithm 2 Modified discrete PSO algorithm.

```
Input: $N$, $m$, $K$, $T$
Output: Best individuals $X_{best}$, fitness function of elite individuals $Y_{best}$
Define parameters: $c_1$, $c_2$, $\Omega_{start}$, $\Omega_{end}$, $\beta$
Initialize $\boldsymbol{x}$ and $\boldsymbol{v}$ of the particle population
Calculate $\boldsymbol{p_i^{best}}$ and $\boldsymbol{g^{best}}$
$i = 1$
while $i \le T$ do
    for $j = 1$ to $N$ do
        Update $\boldsymbol{x}$ and $\boldsymbol{v}$
        if $G(\boldsymbol{x}) \ne 0$ then
            Modify $\boldsymbol{x}$ and $\boldsymbol{v}$
        end if
        Calculate $\boldsymbol{p_i^{best}}$ and $\boldsymbol{g^{best}}$
        Update $X_{best}$ and $Y_{best}$
    end for
    $i = i + 1$
end while
```

We should check whether the constraint is satisfied after the position and velocity of particles are updated in each iteration, and then modify $\boldsymbol{x}$ and $\boldsymbol{v}$ if $G(\boldsymbol{x})$ is not satisfied. If $G(\boldsymbol{x}) < 0$, we randomly choose $G(\boldsymbol{x})$ elements in $\boldsymbol{x}$ that are originally 0, and set them to 1. If $G(\boldsymbol{x}) > 0$, we also randomly choose $G(\boldsymbol{x})$ elements in $\boldsymbol{x}$ that are originally 1, and set them to 0. At the same time, the velocity $\boldsymbol{v}$ on those positions should also be set randomly.

TABLE I
Parameters of the MD-PSO Algorithm

Symbol	Description	Reference	Range
K	Population size	80	$[10, 120]$
T	Maximum iteration number	500	$[200, 1000]$
Ω_{start}	Maximum inertia weight	0.9	$[0.1, 0.9]$
Ω_{end}	Minimum inertia weight	0.4	$[0.1, 0.9]$
c_1	Cognition factor	2	$[0.1, 3]$
c_2	Cognition factor	2	$[0.1, 3]$
β	Critical coefficient	0.5	$[0.1, 0.9]$

TABLE II
Parameters of the GPO Model

Symbol	Description	Value
ϵ	Traffic demand per user	100 Kbps
D_s	Satellite coverage diameter	30°
C_s	Number of gateway antennas	4
B_s	Satellite GSL bandwidth	30 Gbps
$D_{\min}$	Minimum distance	3°
α	Index factor of distance	1
m	Number of existing gateways	30
N	Total number of potential gateways	471

TABLE III
Other Parameters in GPO Model

w_1^o	w_3^o	w_4^o	w_5^o	a	b
1	10^2	1	10^2	1	1

D. Algorithm Analysis

The parameters of the MD-PSO algorithm as listed in Table I can affect the algorithm's performance in terms of the fitness function value and convergence rate, namely the number of iterations. Thus, we first carry out the sensitivity and convergence analyses on those parameters. Furthermore, as K and T can also affect the computational burden, we evaluate the CPU time for computation instead of the number of iterations. Finally, we provide a complexity analysis of the algorithm.

The reference value and range of each parameter (also shown in Table I) are presumed to be potentially good, based on the experiences of previous studies. When we analyze one certain parameter, we fix the others and set them to the reference parameters. In particular, when we analyze the influence of ω, we set it to be a constant value ranging from 0.1 to 0.9, in contrast to ω decreasing linearly from Ω_{start} to Ω_{end} throughout the iterations. When we analyze other parameters, ω is set to linearly decrease as the reference. Besides, parameters c_1 and c_2 are set to be identical, as the self-cognition and social-influence are regarded as equally important in our problem.

With the parameters in Table II, we run the algorithm 10 times independently in MATLAB on a Linux server with 8-core 2.2 GHz Xeon CPU and 32 GB memory. The function weight coefficients or penalty factors are kept constant during our simulation, as in Table III. Here, w_3^o and w_5^o are set to 100 in order to punish the fitness function when $g_1(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$ are not satisfied. We use two quantities to reflect the algorithm's performance, namely the average values of the fitness function $F(\boldsymbol{x})$ and the number of iterations when the best solution is found.

The results of β are shown in Fig. 14, from which we can see that a smaller β can have a better solution for the GPO problem, but it can also increase the times of iterations for reaching the optimal solution, while a bigger β cannot help the algorithm explore the solution domain deeply. It seems that when $\beta = 0.3$, the algorithm can have a good performance with a relatively small number of iterations. The results of cognition factors c_1 and c_2 are depicted in Fig. 15. When c_1 and c_2 are set to less than 1, the fitness function is much bigger and is not plotted here. When their value varies from 1.5 to 2.5, the algorithm has relatively good performance. Moreover, the number of iterations does not change too much when the cognition factors vary from 1 to 3.

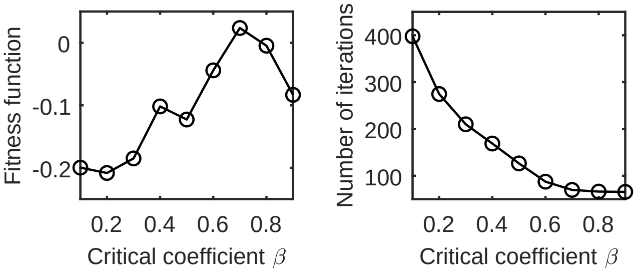


Fig. 14. Sensitivity analysis of the critical coefficient β.

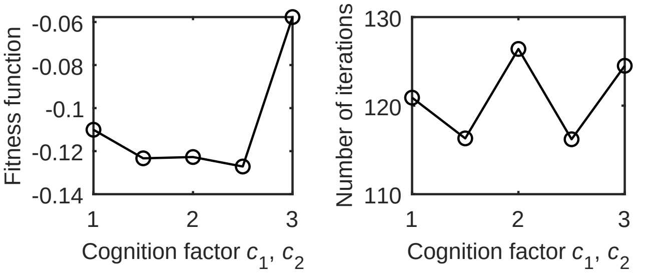


Fig. 15. Sensitivity analysis of cognition factors c_1 and c_2.

When it comes to the inertia weight ω, we can see from Fig. 16 that as ω varies from 0.1 to 0.7, the fitness function is much poorer and the number of iterations is around 50, which means that the algorithm does not have good performance. Compared with the reference case where ω decreases linearly from 0.9 to 0.4, if $\omega = 0.9$, a better solution can be obtained at the expense of doubling the number of iterations.

The impacts of population size K and total number of iterations T are shown in Fig. 17 and 18, where the red dots on the left side represent the fitness function of each independent run, and the black circles are their average values. Fig. 17 shows that even though the average fitness function decreases as the population size increases, the best result for each K has similar value especially when K is more than 80, thus suggesting that $K \geq 80$ is enough. On the other hand, the total

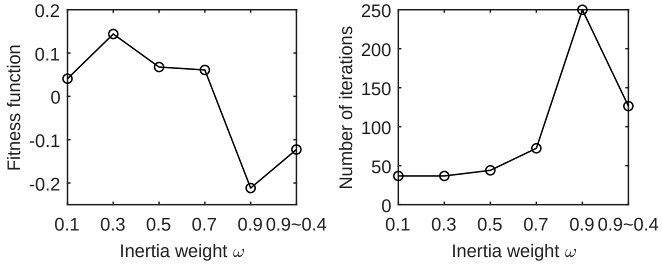


Fig. 16. Sensitivity analysis of inertia weight ω.

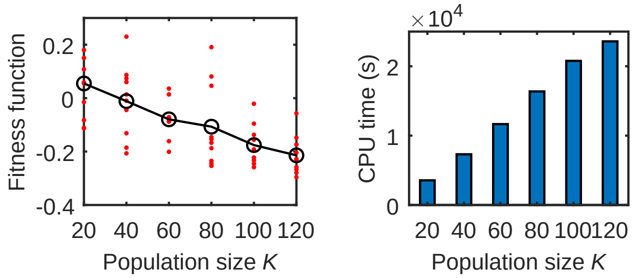


Fig. 17. Sensitivity analysis of population size K.

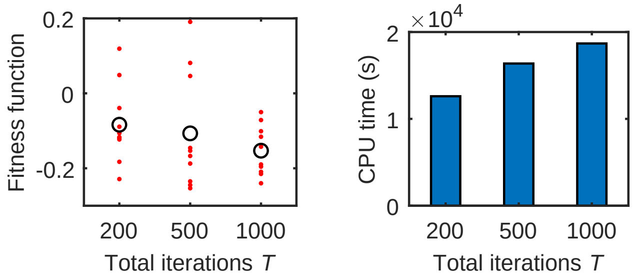


Fig. 18. Sensitivity analysis of total number of iterations T.

TABLE IV
Parameters of Telesat Constellation

Set	P	S	h (km)	i	E
1	6	12	1000.0	99.5°	20.0°
2	5	9	1248.0	37.4°	20.0°

TABLE V
Equivalent parameters of GPO Model for Telesat

ϵ (Kbps)	D_s	C_s	B_s (Gbps)	$D_{\min}$	α	N
100	31.35°	4	20	3°	1	471

CPU time (indicating the computational burden) increases with K, but the increasing rate will slow down if $K \geq 80$, due to a higher convergence rate. According to Fig. 18, T has a small effect on the average value of fitness function, and the best result for each K remains almost constant.

Based on all the above, we choose the PSO parameters that have good performance, and also consider the convergence rate and computational burden in the section covering our case study.

In terms of the computational complexity, we divide the modified discrete PSO algorithm into two parts: the optimizing iteration loop and the calculation of the fitness function. The complexity of the first part is $O(n^2)$ for the worst case, according to [32].

The fitness function consists of six independent steps, and we now analyze their computational complexity one by one. We denote N as the effective gateway number. First, in the process of computing the spherical Voronoi division, the algorithm traverses all the seed points (of the effective gateway number N) to obtain all edges of the spherical polygons. For each of them, there are N vertices in the worst case, and the algorithm calculates the chain of edges. Therefore, the complexity of this step is $O(N^2)$. Second, in the process of computing the attractive/repulsive factor of gateways, the algorithm should traverse all the spherical polygons, thus making a loop of N times. For each time, we need to judge whether the north pole (south pole) is inside the polygon, which causes the coordinate transformation at most N times. After that, the edges of each polygon are traversed to find the cross point with a certain longitude. Hence, the complexity is $O(2N^2)$. Third, in terms of calculating the traffic matrix, the complexity is $O(|K_{pop}|N)$, where the constant value $|K_{pop}|$ is the number of population density units that are non-zero. Fourth, the process of calculating the constraint $g_1(\boldsymbol{x})$ obtains the symmetric distance matrix of each gateway and introduces complexity of $O(N(N-1)/2)$. Fifth, when calculating $g_2(\boldsymbol{x})$, for the worst case, we traverse j (from 2 to the order of the distance matrix of N gateways). For each j, we use a loop of a combinatorial number $C_N^j = N!/((N-j)!j!)$ to calculate the gateways inside a certain satellite coverage area. Thus, the number of calculation is $\sum_{j=2}^{N} C_N^j = 2^N - (N+1)$. As a result, the complexity of this step is $O(2^N)$. Finally, the complexity of calculating the constraint $g_3(\boldsymbol{x})$ is $O(1)$. To conclude, the computational complexity of the fitness function is $O(3N^2 + |K_{pop}|N + N(N-1)/2 + 2^N + 1) = O(2^N)$.

VI. Case Study

A. Telesat Constellation

The Telesat LEO constellation, scheduled to operate in 2022, is used to test the proposed GPO model. Parameters of the Telesat constellation are listed in Table IV, where P is the orbit number, S is the satellite number per orbit, h is the orbit altitude, i is the orbit inclination angle, and E is the minimum ground elevation angle. It adopts a multiple orbit-set configuration with some satellites in inclined orbit and others in polar orbit. This ensures continuous two-fold global coverage [1], such that at least two satellites are within the line of sight for any given location. Each Telesat satellite has two gateway antennas, and its total gateway link bandwidth is 10 Gbps. Nonetheless, C_s and B_s of the Telesat system should be doubled for our simulations because there are always 2 satellites providing services to any terrestrial users. The average value of traffic demand per user is $\epsilon = 100$ Kbps, which means that the actual data rate can sometimes surpass it. In addition, we can calculate the coverage diameter, D_s, of any satellite with (16). Equivalent values of parameters in

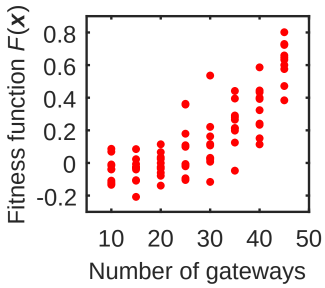


Fig. 19. Fitness function $F(\boldsymbol{x})$ of different number of gateways.

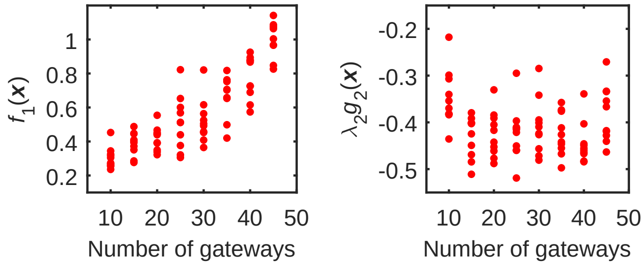


Fig. 20. Objective functions $f_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ of different number of gateways.

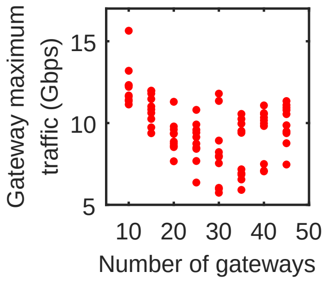


Fig. 21. Gateway maximum traffic varies with number of gateways.

TABLE VI
Experimental set-up for obtaining Pareto front

m	Trial times	w_1^o	w_4^o
28	10	1	1
	10	1.8	0.2
	10	0.2	1.8
29	10	1	1
	10	1.8	0.2
	10	0.2	1.8
30	10	1	1
	10	1.8	0.2
	10	0.2	1.8
31	10	1	1
	10	1.8	0.2
	10	0.2	1.8

GPO model that we adopt here are listed in Table V, except for m, whose value will change in our simulations.

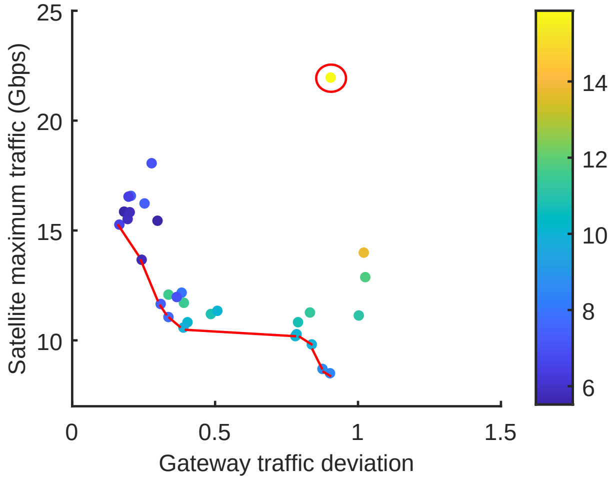
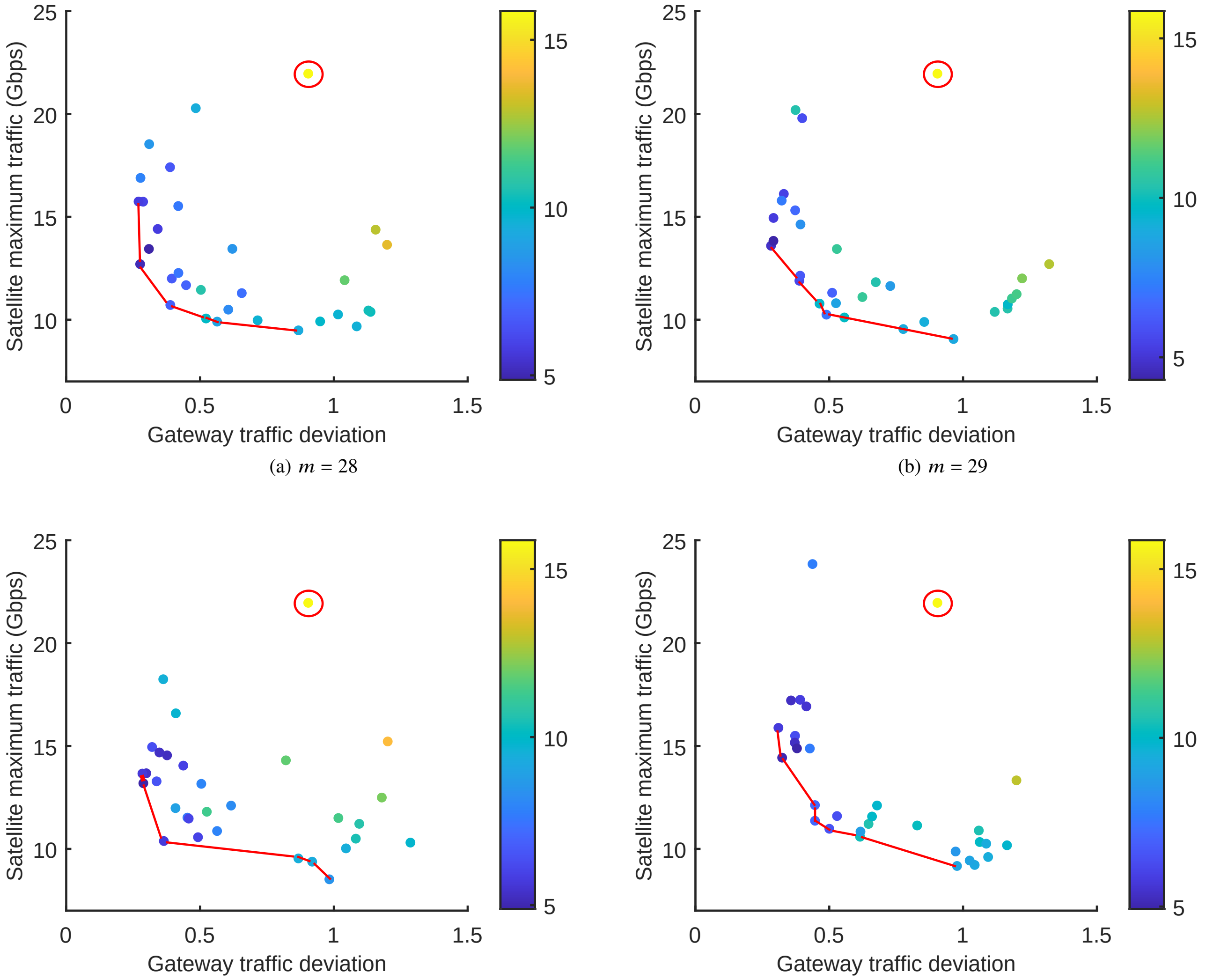


Fig. 22. The approximated Pareto set with 21 gateways. The red circle emphasizes the reference placement scheme.

We run the algorithm 10 times independently for each m, and obtain the fitness function $F(\boldsymbol{x})$ shown in Fig. 19. Generally, $F(\boldsymbol{x})$ can take smaller values with fewer gateways, even though that of the best solution for each m cannot decrease obviously if the number of gateways is fewer than 30. Fig. 20 presents the traffic deviation $f_1(\boldsymbol{x})$ and bandwidth limitation $g_2(\boldsymbol{x})$ of different numbers of gateways. $f_1(\boldsymbol{x})$ can take smaller values when there are fewer gateways, whilst $g_2(\boldsymbol{x})$ has no explicit relationship with the number of gateways. When there are fewer gateways, it is easier to find a load-balancing placement scheme that satisfies the bandwidth constraint; when the number of gateways increases, their traffic load is difficult to balance.

It seems that a placement scheme with fewer gateways can have better results in our GPO model. However, too few gateways may also cause them to have high traffic load. We calculate the gateway's maximum traffic, which is depicted in Fig. 21. We can see that it reaches the lowest value when $m = 30$, and it becomes reasonably larger when $m < 30$. The gateway's maximum traffic increases when $m > 30$, because more gateways can exacerbate the load balancing and increase the gateway's maximum traffic even more.

Accordingly, we choose around 30 gateways for conducting more experiments to obtain the Pareto set, since there is a trade-off between $f_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$. To this end, we change the weight factors, namely w_1^o and w_4^o of $f_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ as it is shown in Table VI. Other weight factors, namely w_3^o and w_5^o, are maintained the same as they are in Table III, because $g_1(\boldsymbol{x})$ and $g_3(\boldsymbol{x})$ only need to be satisfied but not optimized. We run the algorithm 10 times for each set of weight factors, so there are 30 optimal results for each m. As the reference placement scheme has 21 gateways, we also obtain the results of the same gateway number for comparison. Optimization results and the Pareto set are shown in Fig. 22 and 23, where the point inside the red circle is the reference placement scheme, and the red lines are the approximated Pareto front in terms of $f_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$. Besides, each point here represents an optimal solution,



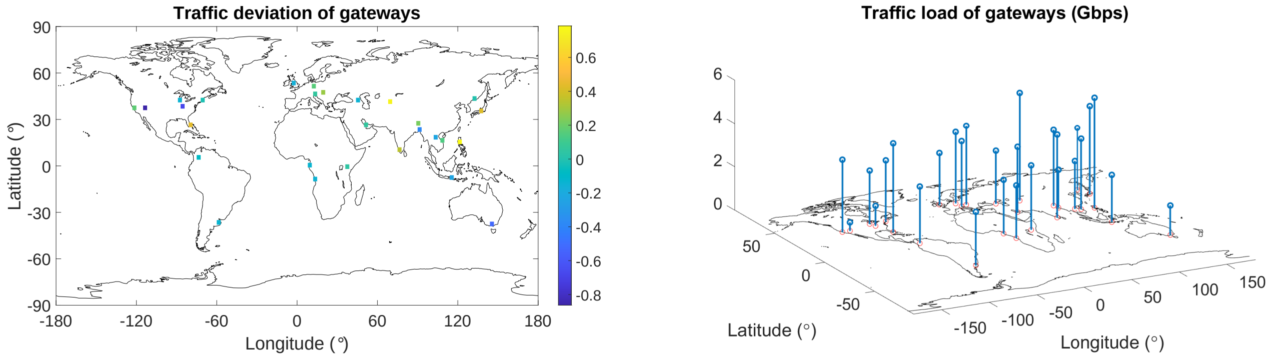
(a) $m = 28$

(b) $m = 29$

(c) $m = 30$

(d) $m = 31$

Fig. 23. The approximated Pareto set with different number of gateways.



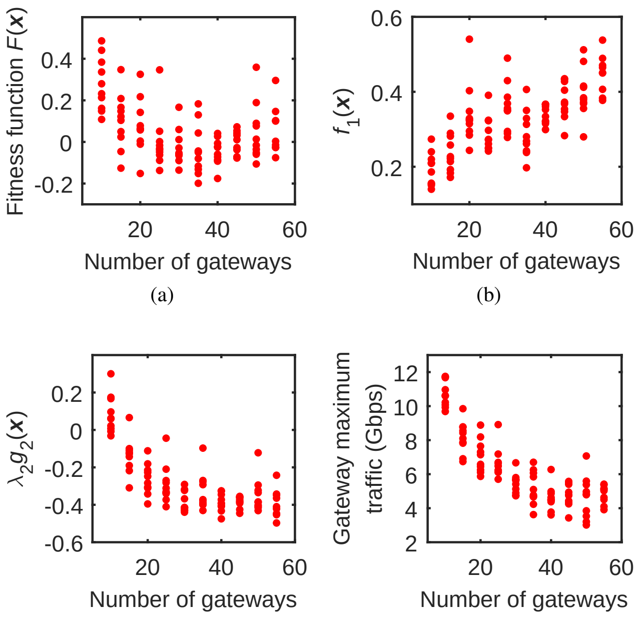
(a) Traffic deviation

(b) Traffic load

Fig. 24. An example of optimal results for $m = 28$.

whose color indicates the gateway's maximum traffic.

In Fig. 22, the optimized solutions show much better performance than the reference placement scheme. The results are aggregated mainly in three regions, namely minimum $f_1(\boldsymbol{x})$, minimum $g_2(\boldsymbol{x})$, and the transition region, which corresponds to the three sets of parameters in Table VI. The minimum



(a)

(b)

(c)

(d)

Fig. 25. Optimization results of OneWeb with different number of gateways.

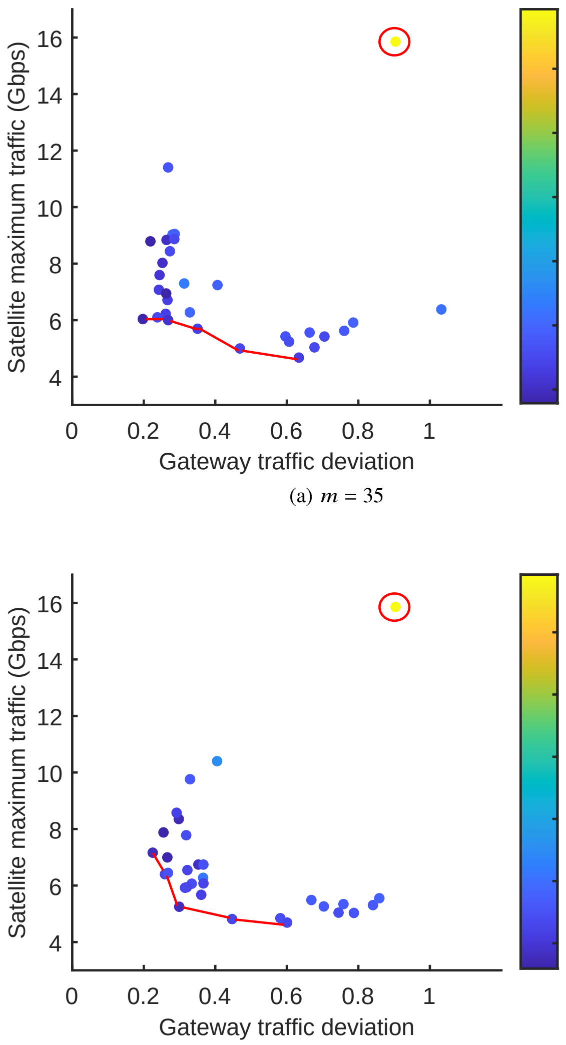
deviation of gateway traffic is about 0.2 with a relatively high satellite's maximum traffic, while it takes the lowest value of about 6 Gbps with larger traffic deviation of gateways.

The results in Fig. 23 have roughly similar profiles, even though they are explicitly affected by the number of gateways. Points closer to the y-axis are darker, which means that the maximum traffic of gateways is mostly lower when their traffic deviation of them is smaller, and it has a large range (5 to 14 Gbps). More blue points exist when more gateways are allowed. In general, fewer gateways can have lower traffic deviation as in (a) and (b), which conforms to results in Fig. 20. Furthermore, points on the approximated Pareto front often have gateway traffic lower than 10 Gbps, regardless of m. Obviously, the performance of our optimization model proves to search deeply, and the optimized solutions are close to the Pareto set.

There is a trade-off between the gateway traffic deviation and the satellite's maximum traffic. Therefore, we choose one solution in Fig. 23 (a), where the traffic deviation is around 0.27 and the satellite's maximum traffic is 13 Gbps, to illustrate the results, which is shown in Fig. 24. More gateways are used in densely-populated areas such as east Asia and Europe, but not in Africa because there are not many IXPs indicating the Internet backbone. On the contrary, a greater number of IXPs results in six gateways in North America, which is quite a large number. Gateways in Asia have the highest traffic load, whilst those in Oceania have the lowest.

B. OneWeb Constellation

We further test the proposed method on the OneWeb constellation of 720 satellites operating in 18 orbit planes with 40 satellites in each orbit. The altitude is 1200.0 km, and the orbit inclination angle is 87.0° with a ground elevation angle of 55.0°. Based on this, we can calculate the parameters of our GPO model for OneWeb. The diameter of the coverage area is 12.27°, and the bandwidth limitation is assumed to be 10 Gbps. Other parameters are the same as in Table V.



(a) m = 35

(b) m = 40

Fig. 26. The approximated Pareto set with different number of gateways.

The simulation results are shown in Fig. 25, and $F(\boldsymbol{x})$ can take better values when m varies from 30 to 40. In terms of $f_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$, the former generally increases with m while the latter decreases with m. Fig. 25 (d) shows that the maximum traffic of gateways has a declining trend with m, but it reaches a stationary phase when $m >= 35$. Based on that, we further calculate the Pareto set of $m = 35$ and $m = 40$, respectively, by changing the weight factor as shown in Table VI. According to the results in Fig. 26, the gateway traffic remains at a low value, as indicated by the dark blue color of most points. Compared with Fig. 23 of Telesat constellation, the y-axis in Fig. 26 has a smaller range, mostly lower than 10 Gbps, which satisfies the bandwidth constraint. Moreover, because the constellations are different, the circled point that represents the same reference placement scheme has different values of satellite's maximum traffic in Fig. 23 and 26.

VII. Conclusion

In this paper, we propose a gateway placement problem in LEO satellite networks, and formulate an optimization model concerning traffic deviation of gateways and satellite communication resources. We adopt the so-called gravity model to estimate the traffic matrix between gateways and users. A discrete PSO method is modified and then used to solve our problem, and sensitivity and convergence analyses are implemented to obtain good parameters of PSO. Finally, we adopt real satellite constellations to obtain the optimal gateway placement scheme and evaluate our optimization model. We obtain solutions to gateway placement of different numbers and find the approximated Pareto front. Results shows that around 30 gateways could be the best choice for the Telesat constellation and 35 to 40 for OneWeb. Our model shows good performance for solving the GPO problem.

Future work will focus on applying our model to other constellations, especially those huge ones with thousands of satellites such as Starlink. In addition, more precise traffic estimation methods could also be taken into consideration for better results.

References

[1] I. d. Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta Astronautica*, vol. 159, pp. 123–135, 2019.

[2] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[3] H. Yao, L. Wang, X. Wang, Z. Lu, and Y. Liu, "The space-terrestrial integrated network: An overview," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 178–185, 2018.

[4] Y. Wang, M. Sheng, W. Zhuang, S. Zhang, N. Zhang, R. Liu, and J. Li, "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 268–279, 2018.

[5] B. Soret and D. Smith, "Autonomous routing for leo satellite constellations with minimum use of inter-plane links," in *2019 IEEE International Conference on Communications (ICC)*, 2019, Conference Proceedings, pp. 1–6.

[6] N. Kato, Z. M. Fadlullah, F. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[7] A. Papa, T. D. Cola, P. Vizarreta, M. He, C. M. Machuca, and W. Kellerer, "Dynamic SDN controller placement in a LEO constellation satellite network," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, Conference Proceedings, pp. 206–212.

[8] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.

[9] Q. Chen, L. Yang, X. Liu, J. Guo, S. Wu, and X. Chen, "Multiple gateway placement in large-scale constellation networks with inter-satellite links," *International Journal of Satellite Communications and Networking*, pp. 1–18, 2020.

[10] M. M. Rahman, S. Saha, U. Chengan, and A. S. Alfa, "IP traffic matrix estimation methods: Comparisons and improvements," in *IEEE International Conference on Communications (ICC)*, vol. 1, 2006, Conference Proceedings, pp. 90–96.

[11] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 206–217, 2003.

[12] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2127–2136, 2006.

[13] J. Robinson, M. Uysal, R. Swaminathan, and E. Knightly, "Adding capacity points to a wireless mesh network using local search," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, Conference Proceedings, pp. 1247–1255.

[14] U. Ashraf, "Energy-aware gateway placement in green wireless mesh networks," *IEEE Communications Letters*, vol. 21, no. 1, pp. 156–159, 2017.

[15] I. Bisio and M. Marchese, "Efficient satellite-based sensor networks for information retrieval," *IEEE Systems Journal*, vol. 2, no. 4, pp. 464–475, 2008.

[16] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, "Efficient gateways placement for internet of things with QoS constraints," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, Conference Proceedings, pp. 1–6.

[17] Y. Cao, H. Guo, J. Liu, and N. Kato, "Optimal satellite gateway placement in space-ground integrated networks," *IEEE Network*, vol. 32, no. 5, pp. 32–37, 2018.

[18] Y. Cao, Y. Shi, J. Liu, and N. Kato, "Optimal satellite gateway placement in space-ground integrated network for latency minimization with reliability guarantee," *IEEE Wireless Communications Letters*, vol. 7, no. 2, pp. 174–177, 2018.

[19] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 221–232, 2018.

[20] K. Yang, B. Zhang, and D. Guo, "Controller and gateway partition placement in SDN-enabled integrated satellite-terrestrial network," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, Conference Proceedings, pp. 1–6.

[21] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in measuring backbone traffic variability: models, metrics, measurements and meaning," in *2nd ACM SIGCOMM Workshop on Internet measurment*. Association for Computing Machinery, 2002, Conference Proceedings, pp. 91–92.

[22] A. Medina, N. Tafta, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *Acm Sigcomm Computer Communication Review*, vol. 32, no. 4, pp. 161–174, 2002.

[23] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone: a comparison on real data," in *ACM SIGCOMM Conference on Internet Measurement*. Association for Computing Machinery, 2004, Conference Proceedings, pp. 149–160.

[24] C. Chen and E. Ekici, "A routing protocol for hierarchical LEO/MEO satellite IP networks," *Wireless Networks*, vol. 11, no. 4, pp. 507–521, 2005.

[25] Y. Yang, M. Xu, D. Wang, and Y. Wang, "Towards energy-efficient routing in satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3869–3886, 2016.

[26] S. Wu, X. Chen, L. Yang, C. Fan, and Y. Zhao, "Dynamic and static controller placement in software-defined satellite networking," *Acta Astronautica*, vol. 152, pp. 49–58, 2018.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, Conference Proceedings, pp. 1942–1948.

[28] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, 1997, Conference Proceedings, pp. 4104–4108.

[29] B. Al-kazemi and C. K. Mohan, "Multi-phase generalization of the particle swarm optimization algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 1, 2002, Conference Proceedings, pp. 489–494.

[30] G. Pampara, N. Franken, and A. P. Engelbrecht, "Combining particle swarm optimisation with angle modulation to solve binary problems," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, Conference Proceedings, pp. 89–96 Vol.1.

[31] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299–308, 2008.

[32] W. Chen, J. Zhang, H. S. H. Chung, W. Zhong, W. Wu, and Y. Shi, "A novel set-based particle swarm optimization method for discrete opti-

mization problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 278–300, 2010.

[33] W. Chen and D. Tan, "Set-based discrete particle swarm optimization and its applications: a survey," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 203–216, 2018.

[34] M. Shen, Z. Zhan, W. Chen, Y. Gong, J. Zhang, and Y. Li, "Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7141–7151, 2014.

[35] I. d. Portillo, B. Cameron, and E. Crawley, "Ground segment architectures for large LEO constellations with feeder links in EHF-bands," in *2018 IEEE Aerospace Conference*, 2018, Conference Proceedings, pp. 1–14.

[36] X. Chen, W. Yao, Y. Zhao, X. Chen, and X. Zheng, "A practical satellite layout optimization design approach based on enhanced finite-circle method," *Structural and Multidisciplinary Optimization*, vol. 58, no. 6, pp. 2635–2653, 2018.

[37] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger, "Peering at peerings: On the role of IXP route servers," in *2014 Conference on Internet Measurement Conference*. Association for Computing Machinery, 2014, Conference Proceedings, pp. 31–44.

[38] PCH, "Internet exchange point datasets," 2019. [Online]. Available: https://www.pch.net/ixp/data

[39] "Gridded population of the world, version 4 (gpwv4): Population count, revision 11," 20200204 2018. [Online]. Available: https://doi.org/10.7927/H4JW8BX5

[40] B. Luong, "Voronoi sphere," 2013. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/40989-voronoi-sphere

[41] M. Bevis and J.-L. Chatelain, "Locating a point on a spherical surface relative to a spherical polygon of arbitrary shape," *Mathematical geology*, vol. 21, no. 8, pp. 811–828, 1989.

Jianming Guo received a BS degree in Astronomy from the University of Science and Technology of China, Hefei, China in 2015, and an ME degree in Aeronautical and Astronautical Science and Technology from the National University of Defense Technology, Changsha, China in 2017. He is currently pursuing a PhD at the College of Aerospace Science and Engineering, National University of Defense Technology. He has been a visiting PhD student at Universitat Politècnica de Catalunya (UPC) since December 2019. His current research interests include SDN, NFV, and satellite communication networks.

David Rincón received an MSc in telecommunication engineering and a PhD in Computer Networks from Universitat Politècnica de Catalunya Barcelona Tech (UPC). In 1998, he joined the Department of Telematics Engineering at UPC, where he is currently an Associate Professor. He has been a visiting researcher at the Teletraffic Research Centre (University of Adelaide, Australia, 2007) and at the Institute of Pure and Applied Mathematics (IPAM) at UCLA (2008). His interests include traffic modeling, network softwarization, optical access networks, and energy consumption in computer networks.

Sebastià Sallent received an MSc and PhD in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona. From 1979 to 1985, he worked for the Philips Company. In 1985, he received a fellowship at the University of Lausanne and in 2009 another at the University of California, Davis. He has been the Director of the i2CAT Internet Research Center and he was president of the Spanish Telematic Association. Currently, he holds a full-time position as professor in the Department of Network Engineering, UPC. He leads the Broadband Networks research group in the same department. His current research interests include access networks, network and service virtualization (SDN/NFV), 5G, and new Internet architectures.

Lei Yang received a PhD from the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China in 2008. He is now a professor of the College of Aerospace Science and Engineering, National University of Defense Technology. Prof. Yang is currently a member of the Chinese Society of Astronautics and the China Instrument and Control Society. His current research interests focus on satellite communication networks, measurement and control technology for micro satellites, on-board computers, spacecraft system modeling, and simulation.

Xiaoqian Chen received his MS. and PhD degrees in aerospace engineering from the National University of Defense Technology, China, in 1997 and 2001, respectively. He is currently a professor at the National University of Defense Technology. His current research interests include spacecraft systems engineering, advanced digital design methods for space systems, and multidisciplinary design optimization.

Xianqi Chen received a BE degree in Aerospace Engineering and an ME degree in Aeronautical and Astronautical Science and Technology from the National University of Defense Technology, Changsha, China, in 2016 and 2018, respectively. He is currently pursuing a PhD in the College of Aerospace Science and Engineering, National University of Defense Technology. His main research interests include intelligent design, multidisciplinary design optimization, satellite layout optimization design and evolutionary algorithms in aerospace engineering.